

03P00098



⑬ **BUNDESREPUBLIK  
DEUTSCHLAND**



**DEUTSCHES  
PATENT- UND  
MARKENAMT**

⑫ **Übersetzung der  
europäischen Patentschrift**

⑨⑦ **EP 0718 727 B 1**

⑩ **DE 695 15 011 T 2**

⑤① Int. Cl. 7:  
**G 05 B 19/042**  
G 05 B 19/05

- ②① Deutsches Aktenzeichen: 695 15 011.1
- ⑨⑥ Europäisches Aktenzeichen: 95 107 119.0
- ⑨⑥ Europäischer Anmeldetag: 11. 5. 1995
- ⑨⑦ Erstveröffentlichung durch das EPA: 26. 6. 1996
- ⑨⑦ Veröffentlichungstag  
der Patenterteilung beim EPA: 9. 2. 2000
- ④⑦ Veröffentlichungstag im Patentblatt: 14. 9. 2000

③⑩ Unionspriorität:  
360862 21. 12. 1994 US

⑦③ Patentinhaber:  
Allen-Bradley Co., Inc., Milwaukee, Wis., US

⑦④ Vertreter:  
Patent- und Rechtsanwälte Holtz, Martin, Lippert,  
Frankfurt, München, 60322 Frankfurt

⑧④ Benannte Vertragsstaaten:  
DE, FR, GB

⑦② Erfinder:  
Hodorowski, John, Hales Corners, US; Pieronek,  
Donald, Brookfield, US

⑤④ Industrielle Steuerungen mit hochverteilter Verarbeitung und Verfahren zu deren Programmierung

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

**DE 695 15 011 T 2**

**DE 695 15 011 T 2**

695 15 011.1-08  
EP 95 107 119.0  
Allen-Bradley Company, Inc.  
91DE

## Industrielle Steuerungen mit hochverteilter Verarbeitung und Verfahren zu deren Programmierung

### Hintergrund der Erfindung

#### 1. Anwendungsgebiet der Erfindung

Die Erfindung betrifft industrielle Steuerungssysteme, einschließlich programmierbare Steuerungen, und insbesondere ein hochverteilter industrielles Steuerungssystem sowie ein Verfahren zu dessen Programmierung.

#### 2. Zugrundeliegender Stand der Technik

Bei industriellen Steuerungen, wie sie beispielsweise in den US-Patenten 3 810 118, 3 942 158, 4 165 534 und 4 442 504 beschrieben sind, handelt es sich typischerweise um zentralisierte Hochgeschwindigkeitscomputer, die mit industriellen Anlagen verbunden werden können, beispielsweise mit automatisierten Montagestraßen oder Werkzeugmaschinen, um derartige Anlagen in Übereinstimmung mit einem gespeicherten Programm zu betreiben. Das gespeicherte Programm enthält Anweisungen, die bei ihrer Ausführung den Zustand ausgewählter Eingänge überprüfen, die von Sensoreinrichtungen an der gesteuerten Anlage der Steuerung zugeführt werden, und ausgewählte Ausgänge oder Aktoren der Steuerung einschalten oder ausschalten, um Einrichtungen an der gesteuerten Anlage zu betreiben.

Eingänge zur industriellen Steuerung können diskrete binäre Signale sein, wie solche von Schaltern, die Grenzen von Prozeßvariablen erfassen können, wie beispielsweise Bewegung, Temperatur, Zeit oder andere Größen, oder bei den Eingängen kann es sich um analoge Messungen von Prozeßvariablen handeln, die dann im allgemeinen zur Verarbeitung in digitale Wörter umgesetzt werden, oder bei den Eingängen kann es sich um multivariable Information handeln.

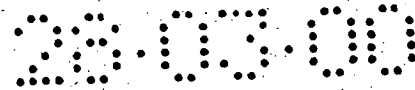


Gleichermaßen können die Ausgänge der industriellen Steuerung entweder diskrete binäre Signale sein, wie sie typischerweise von mechanischen oder Festkörperrelais erzeugt werden, oder analoge Ausgänge, die mit Hilfe eines Digital/Analog-Umsetzers erzeugt werden, oder multivariable Befehle. Einige der Eingänge und Ausgänge der zentralisierten Steuerung können entfernt angeordnet und mit der Steuerung über eine digitale Kommunikationsverbindung verbunden sein. Typischerweise verbindet das Netzwerk die Steuerung mit einem Fern-E/A-Gestell, bei dem eine Anzahl Eingänge und Ausgänge angehäuft sind.

Die zentralisierte Architektur von derzeitigen industriellen Steuerungssystemen spiegelt sowohl die historisch hohen Kosten von Computerhardware (die durch die Verwendung einer einzigen zentralisierten Steuerung vermindert werden könnte) wieder als auch den Wunsch für ein zentralisiertes Berichtswesen und eine zentralisierte Koordination des Betriebs eines ganzen industriellen Prozesses. Dennoch ist eine zentralisierte Steuerungsarchitektur für alle Steuerungsaufgaben nicht ideal.

Für einige einfache Steuerungsaufgaben mit wenigen Eingängen und Ausgängen kann die zentralisierte Steuerung unnötig kompliziert und aufwendig werden und unnötige Eigenschaften und Kapazitäten haben. Umgekehrt können für einige komplexe Steuerungsaufgaben mit vielen Eingängen und Ausgängen die Geschwindigkeitsgrenzen der zentralisierten Steuerung oder ihre Verbindungen zu Fern-Eingängen und -Ausgängen die Leistung des Steuerungssystems beträchtlich vermindern. Die zentralisierte Steuerung verarbeitet jeden Eingang und Ausgang des Steuerungssystems sequentiell. Bei großen Programmen kann eine beachtliche Verzögerung zwischen der Änderung eines Eingangs und der Steuerung eines entsprechenden Ausganges auftreten. Ist die Verzögerung zu groß, kann das Steuerungssystem träge oder unvorhersagbar werden. Bei einer zentralisierten Steuerung müssen der Zustand jedes Eingangs und Ausganges wiederholt zu der zentralisierten Steuerung mit einer Geschwindigkeit übertragen werden, die für die Zielerreichung hinreichend ist. Sind viele Fern-Eingänge und -Ausgänge des Steuerungssystems vorhanden, kann die Kapazität der Verbindung zwischen der E/A und der zentralisierten Steuerung eine beachtliche Einschränkung darstellen.

In einer zentralisierten Architektur setzt ein Ausfall der einzigen Steuerung typischerweise das gesamte Steuerungssystem lahm. Eine Fehlersuche nach dem Grund des Ausfalls der Steuerung wird behindert, wenn der Fehler oder die Störung von einer solchen Natur ist,



daß die ausgefallene Steuerung keine Information über ihren internen Betrieb melden oder fehlersuchende Programme ausführen kann.

Derzeitige industrielle Steuerungen sind gewöhnlicherweise mittels einer "Leiter-logik"- oder Kontaktplansprache programmiert, in der das Steuerprogramm in Form von Leitern dargestellt ist, deren vertikale Schienen eine Energiequelle oder Energiesenke und deren Sprossen seriell und parallel miteinander verbundene Kontakte und Relaispulen darstellen. Die Kontakte können entweder normalerweise offen oder normalerweise geschlossen sein, und entweder durch externe Eingänge oder Relaispulen gesteuert werden. Der begrenzte Satz an Befehlen in einem Kontaktplan (Kontakte und Spulen) kann in effizienter Weise durch einen Computer mit reduziertem Befehlsvorrat ("RISC") verarbeitet werden.

Ogleich diese Sprache vom Konzept her einfach ist kann es bei extrem langen Programmen für einen Programmierer schwierig sein, einen Kontaktplan zu interpretieren. Das Aufsuchen von Fehlern kann in großen Programmen, die in der Leiter-Logik geschrieben sind, schwierig sein, da man in einem solchen Leiter- oder Kontaktplan den Steuerungsfluß nicht leicht erkennen kann. Weiterhin benötigt man für die meisten Steuerungssysteme über Kontakte und Relaispulen hinausgehende zusätzliche Funktionselemente, wie beispielsweise Zähler, Vergleicher, Zeitgeber und dergleichen, deren Implementation als Relaislogik äußerst mühsam ist.

Die "Dritte Internationale Konferenz betreffend Softwareentwicklungen in Echtzeitsystemen" (Conf. Publ. No. 344), Cirencester, UK, 16. - 18. September 1991, ISBN 0-85296-526-5, 1991, London, UK, IEE, UK, Seiten 139 - 144, Juer, J. et al: "IEC65A control language - a practical view", behandelt die Vorzüge und Grenzen des IEC65A-Standards. Eine der darin genannten Beschränkungen besteht darin, daß die Kommunikation zwischen Ressourcen auf globale Variable oder nominierte "Zugriffsvariable" beschränkt ist, und zwar unter Verwendung von Sende- und Empfangsfunktionsblöcken. Um von einem größeren Bereich von Daten von einer Ressource zu einer anderen Ressource Gebrauch zu machen, offenbart das oben erwähnte Dokument einen Mechanismus, der es einem Benutzer ermöglicht, beim Schreiben in IEC65A Sprachen, die eine zentrale Konfiguration benötigen, Ferndaten so zu gruppieren, als ob es sich dabei um einen lokalen Funktionsblock handeln würde. Dieser Mechanismus macht die verteilte Programmierung unter Verwendung von IEC65A-Standards viel leichter als die Verwendung von Kommunikationsblöcken. In diesem Zusammenhang



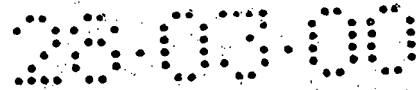
wird die Konfiguration als eine Spezifikation darüber betrachtet, wo am Netzwerk Ressourcen platziert sind. Die Ressourcen sind mit einem Stück physischer Hardware assoziiert, und sie werden umschrieben durch einen Satz miteinander in Beziehung stehender textueller und graphischer Programmiersprachen, d. h. durch Softwareblöcke wie beispielsweise Funktionsblöcke. Irgendwelche Verbindungslinien, die zwischen den Ressourcen oder deren Blöcke erscheinen, sind Verbindungen zwischen Softwareblöcken.

Elektronik, 29. September 1992, DE, Band 41, Nr. 20, ISSN 0013-5658, Seiten 110 - 115, Süß, G.: "Generationswechsel bei SPS-Entwicklungssystemen", offenbart textuelle und graphische Programmiersprachen der Norm IEC 1131-3 zur Definition von Programmelementen wie Funktionen, Funktionsblöcke und Programmen. Eine dieser graphischen Programmiersprachen ist der Kontaktplan (ladder diagram), der aus Kontakten und Relaispulen besteht, um innerhalb eines von einem Prozessor auszuführenden Steuerprogramms logische Funktionen zu definieren. Wie bereits oben erwähnt, ist allerdings die Kontaktplansprache zu beschränkt, so daß für die meisten Steuerungssysteme zusätzliche Funktionselemente erforderlich sind, die über Kontakte und Relaispulen hinausgehen.

I & CS - Industrial and process Control Magazine, Vol. 64, No. 10, 1. Oktober 1991, Seite 146, Labs, W.: "Windows Software combines Modelling, PLC code generation", offenbart ein Softwarepaket, das auf der Grundlage visueller symbolischer Diagramme Programmier- und Simulationsfähigkeiten vorsieht. Zusätzlich verwendet die Software sequentielle Funktionsdarstellungen und Relaisleiterlogik. Wie die Programmiersprache IEC65A und IEC 1131-3 ermöglicht es die in diesem Aufsatz beschriebene Software dadurch Programme zu schreiben, das standardmäßige sequentielle Funktionsdarstellungssymbole oder Relaisleiterlogiksymbole miteinander verbunden werden, d. h. Symbole von Programmblöcken oder Softwareelementen. Zusätzlich erzeugt die offenbarte Software eine Laufsimulation des tatsächlichen Steuerungssystems, beispielsweise einer Produktionsstraße. Der Code zur Simulation ist derselbe Code, der die programmierbaren Steuerungen zur Steuerung der Produktionsstraße laufen läßt.

### **Kurze Zusammenfassung der Erfindung**

Die Erfindung ist ein Architektur- und Programmierverfahren für ein hochverteiltes industrielles Steuerungssystem, bei dem jeder der Steuerungsmodule, beispielsweise jeder



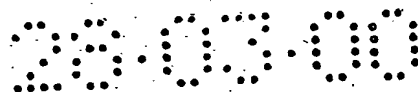
Eingang, Sensor, Ausgang oder Aktor, einen kleinen Teil des gesamten Steuerprogramms trägt. Die verteilte Natur dieser Architektur gestattet extrem schnelle Verarbeitungszeiten, da das aktuelle Programm auf verschiedenartigen Steuerungsmodulen gleichzeitig ausgeführt werden kann. Die verteilte Natur der Architektur verbessert auch die Chance, daß ein Ausfall oder eine Störung einer einzigen Komponente das gesamte Steuerungssystem lahmlegt, und es verbessert die Fähigkeit der Suche nach irgendeinem Fehler oder irgendeiner Störung dadurch, daß die verbliebenen, nicht geschädigten Steuerungsmodule abgefragt werden.

Wichtig ist, daß die Erfindung eine hochverteilte Steuerungsarchitektur vorsieht, bei der jedes Steuerungsmodul eine standardmäßige elementare elektrische Komponente nachbildet, wie beispielsweise einen Schalter oder eine Lichtquelle. Dieser Vorschlag sieht nicht nur eine Architektur vor, sondern auch eine "Programmiersprache" in Form eines elektrischen Schemabildes, das insbesondere für einfache industrielle Steuerungen geeignet ist.

Aus Effizienzgründen können mehr als eine elementare Funktion von einem gegebenen elektronischen Modul ausgeführt werden. So kann beispielsweise eine Vielzahl Drucktastenschalter, von denen jeder eine elementare Funktion darstellt, in einem einzigen Gehäuse mit einem einzigen Mikroprozessor untergebracht sein, der mit der gemeinsamen Verbindung kommuniziert. Weiterhin haben einige elementare Funktionen, beispielsweise ein Zeitgeber, keine Verbindung zur Außenwelt und können deshalb in effizienter Weise durch ein elektronisches Modul nachgebildet werden, das eine andere Haupttask hat, beispielsweise diejenige der Steuerung einer Lichtquelle.

Eine Feldplanungszeichnung sieht ein einfaches Verfahren vor, um Stücke des gesamten Steuerprogramms auf die verschiedenartigen Steuerungsmodule aufzuteilen - ein Problem, das bei dem Versuch einer "Parallelverarbeitung" oft Ärger bereitet. Die Feldplanungszeichnung definiert flexibel und intuitiv, welche physische Hardware welche elementaren Funktionen nachbildet.

Die Erfindung sieht insbesondere eine grafische Sprache zum Programmieren einer hochverteilten industriellen Steuerung vor, die eine große Anzahl physisch getrennter Module enthält, von denen jedes elementare elektrische Funktionen nachbildet, wobei die Module längs eines gemeinsamen Kommunikationsmittels durch Aussenden von Nachrichten auf



dieses Mittel kommunizieren und wobei ein Modul einige elementare Funktionen nachbilden kann.

Die Programmierung umfaßt zunächst das Generieren eines logischen Diagramms, das die Industriesteuerung in der Form eines elektrischen schematischen Bildes aus elektrischen Symbolen auf einem elektronischen Displayschirm darstellt. Die Symbole repräsentieren jeweils eine elementare elektronische Funktion und haben wenigstens einen Eingangsanschluß und einen Ausgangsanschluß, zwischen denen gemäß der elementaren Funktion Strom fließt. Die Anschlüsse eines Symbols sind auf dem schematischen Bild mit den Anschlüssen von anderen Symbolen verbunden, und wenigstens ein elektrisches Symbol ist mit einem Energieversorgungssymbol verbunden. Beim zweiten Schritt der Programmierung wird ein physisches Diagramm generiert, das Umrisse darstellt, die tatsächliche Module repräsentieren und mit Symbolen verbunden sind, die die von den Modulen auszuführenden elementaren Funktionen darstellen. Die Module werden programmiert, um die elementaren Funktionen nachzubilden, die ihnen durch das physische Diagramm zugeordnet sind, und die elementaren Funktionen werden virtuell gemäß dem logischen Diagramm des schematischen Bildes verbunden. Der letztere Schritt wird dadurch ausgeführt, daß das physische und das logische Diagramm zum Generieren einer Kommunikationstabelle kompiliert werden. Die Kommunikationstabelle erstellt wiederum das Kommunikationsprotokoll zwischen den Modulen am Kommunikationsmittel, so daß die von jedem Modul ausgeführten elementaren Funktionen mit ausgewählten anderen elementaren Funktionen so kommunizieren können, als ob diese elementaren Funktionen gemäß dem schematischen Bild miteinander verdrahtet wären.

Die Aufgabe, die durch die im Anspruch 1 definierte Erfindung gelöst werden soll, besteht somit darin, ein Verfahren zum Programmieren hochverteilter Industriesteuerungen vorzusehen, das einen intuitiven Weg anbietet, um nicht nur die Logik der Steuerung zu beschreiben, sondern auch die Verarbeitungsfunktionen verschiedenartigen räumlich verteilten Modulen zuzuteilen.

Beim Entwurf industrieller Prozesse werden derzeit sowohl logische Schemabilder als auch physische "panel-layout"-Ansichten hergestellt. Diese Zeichnungen werden benutzt, um die Konstruktion und Betriebsweise diskreter Industriesteuerungssysteme für Elektriker und andere, die die verschiedenartigen Komponenten anordnen und miteinander verdrahten müssen, beschreiben. Die schematischen Darstellungen und panel-layout-Zeichnungen werden

typischerweise auf computerunterstützten Konstruktionssystemen ("CAD") hergestellt. Die Erfindung benutzt Datendateien, die vom CAD-System als Eingang zu einem Compiler entwickelt worden sind, der die Funktionen unter den Modulen zuteilt und sie virtuell über Nachrichten auf der Verbindung miteinander verdrahtet.

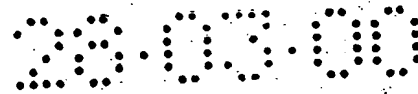
Ein weiteres Ziel der Erfindung ist es somit, eine Programmiersprache vorzusehen, die möglichst geringe Anforderungen an den Konstrukteur industrieller Steuerungen stellt. Die Verwendung CAD-hergestellter panel-layout-Zeichnungen zum Zuteilen der Verarbeitungsfunktionen zwischen den verteilten Komponenten fügt dem Konstruktionsprozeß keine zusätzlichen Schritte hinzu.

Jedes Modul kann einen vorbestimmten Fähigkeitsindex haben, der ihm zugeordnet ist und diejenigen elementaren Funktionen anzeigt, die dieses Modul ausführen kann, oder wieviele einer besonderen Art elementarer Funktionen von diesem Modul ausgeführt werden können. Zuordnungen der elementaren Funktionen des logischen Diagramms zu einem besonderen Modul können unter Bezugnahme auf den Fähigkeitsindex überprüft werden, um dem Programmierer einen Warnhinweis zu geben oder um alternativ die elementare Funktion einem anderen Modul zuzuordnen. Die Zuordnung oder Zuteilung von elementaren Funktionen zu Modulen kann derart beschränkt werden, daß elementare Funktionen, die einem besonderen Modul zuzuordnen sind, beispielsweise wegen Verbindungen zur Außenwelt, nicht verschoben werden.

Es ist deshalb ein weiteres Ziel der Erfindung, eine automatische Zuordnung elementarer Funktionen zu räumlich getrennten Modulen gemäß dem vom Konstrukteur erstellten physischen Diagramm zu gestatten, allerdings in einer Weise, die mit den physischen Grenzen der elektronischen Module in Einklang steht. Macht man von einem Fähigkeitsindex Gebrauch, dann gestattet dieser eine geeignete Zuordnung, die ohne Eingriff der Bedienungsperson und ohne Beeinträchtigung des schematischen Diagramms verifiziert ist.

Die vorstehenden und andere Ziele und Vorteile der Erfindung gehen aus der nachfolgenden Beschreibung hervor. In dieser Beschreibung wird auf die anliegenden Zeichnungen bezug genommen, die ein Teil der Beschreibung bilden und in denen beispielshalber ein bevorzugtes Ausführungsbeispiel der Erfindung gezeigt ist. Aus Gründen der Klarheit sind zwei





einfache Beispiele eines Steuerungssystems in Betracht gezogen. Solche Ausführungsbeispiele stellen notwendigerweise nicht den vollen Umfang der beanspruchten Erfindung dar.

### Kurze Beschreibung der Zeichnungen

Fig. 1 ist eine bildliche Darstellung eines einfachen, hochverteilten Steuerungssystems nach der Erfindung mit einer Anzahl miteinander verbundener Steuerungsmodule, die mittels eines Programmiergeräts zur Steuerung eines Motors für ein Fördersystem zu programmieren sind;

Fig. 2 ist eine vereinfachte Darstellung eines Fördersystems, das durch das Steuerungssystem von Fig. 1 zu steuern ist;

Fig. 3 ist eine schematische Darstellung des Steuerungssystems von Fig. 1, und zwar in einer Implementation mit standardmäßigen elektrischen Komponenten, die die Aufteilung der Funktionalität dieser Komponenten in die Steuerungsmodule von Fig. 1 zeigen;

Fig. 4 ist ein schematisches Diagramm eines Steuerungsmoduls, der zur Verwendung in der Erfindung geeignet ist;

Fig. 5 (a) bis (d) sind beispielhafte Sätze mit Anschlüssen versehener, elektrischer grafischer Symbole, die die elementaren Funktionen der Teile der Steuerungsmodule der Erfindung definieren können und die zusammen mit Drahtverbindungsleitungen die Programmiersprache des hochverteilten Steuerungssystems der Erfindung bilden;

Fig. 6 ist eine vereinfachte Darstellung der Steuerungsmodule von Fig. 1 und zeigt ihre Adressen und die Inhalte ihrer Verbindungslisten, die die virtuelle Verbindung zwischen den Steuerungsmodulen vorsehen;

Fig. 7 ist eine mehr vervollständigte Darstellung des Speichers in jedem Steuerungsmodul und zeigt sowohl die Verbindungslisten von Fig. 6 zusammen mit anderen Daten, die die Programmierung definieren, als auch logische Attribute der Steuerungsmodule;

Fig. 8 ist ein Flußdiagramm der Betriebsweise des Programmiergeräts von Fig. 1, wie es zur Programmierung des Steuerungssystems sowie zum Zuordnen und Herabladen dieses Programms zu den verschiedenartigen Modulen benutzt wird;

Fig. 9 ist ein Flußdiagramm des Betriebs eines Steuerungsmoduls von Fig. 1 und zeigt dessen Verhalten auf verschiedenartige Nachrichten am Netzwerk einschließlich derjenigen von anderen Modulen und vom Programmiergerät;

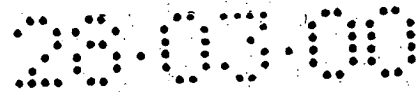


Fig. 10 ist ein Bildschirmdisplay, wie man es auf dem Programmiergerät von Fig. 1 sehen kann, und zeigt ein Teilemagazin aus Steuerungsmodulen und Teilen von Steuerungsmodulen, wie man sie in Fig. 5 findet, und verschiedenartige Programmierertools, die man zum Generieren eines schematischen Bildes verwenden kann;

Fig. 11 ist die vervollständigte schematische Darstellung der Steuerungslogik für die hochverteilte Industriesteuerung von Fig. 1 und 3, wie sie auf dem Schirm von Fig. 10 dargestellt ist und wie sie benutzt wird, um die notwendigen Verbindungslisten und Verdrahtungslisten für das Steuerungssystem zu erzeugen;

Fig. 12 ist eine perspektivische Ansicht eines einfachen Prozesses zum Stapeln und Freigeben von Kugeln, wie er in dem hochverteilten Steuerungssystem der Erfindung verwendet werden kann;

Fig. 13 ist eine vervollständigte schematische Darstellung eines alternativen Industriesteuerungssystems, das aus elementaren Funktionen gebildet ist, wie beispielsweise aus Schaltern und Solenoiden, die mit Drähten miteinander verbunden sind, wobei diese schematische Darstellung auf einem CAD-System entwickelt werden kann und eine logische Beschreibung der Industriesteuerung darstellt;

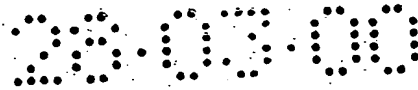
Fig. 14 ist eine physische Ansicht oder ein "panel layout" der Industriesteuerung von Fig. 13 und zeigt Darstellungen der tatsächlichen Steuerungsmodule, die zur Implementierung der Industriesteuerung verwendet werden, einschließlich physischer Symbole der verschiedenen elementaren Funktionen, die in Fig. 13 dargestellt sind;

Fig. 15 ist ein Flußdiagramm des Kompilierungsprozesses der Erfindung, bei dem die logischen und physischen Ansichten von Fig. 12 und 13, wie sie auf einem elektronischen Displaygerät dargestellt und durch grafische Koordinaten repräsentiert sind, zu Listen kompiliert werden, die vereint werden können, um eine Verbindungstabelle zu erzeugen, die zur Kommunikation der physischen Module von Fig. 14 miteinander benutzt wird;

Fig. 16 ist eine grafische Darstellung einer die elementaren Funktionen betreffende Datenbank und zeigt Zusatzinformationen über die elementaren Funktionen, die in dem logischen Diagramm von Fig. 13 miteinander verbunden werden können;

Fig. 17 ist eine grafische Darstellung einer die elektronischen Module betreffende Datenbank und zeigt Informationen über die physischen Module, die in dem logischen Diagramm von Fig. 13 miteinander verbunden werden können; und

Fig. 18 ist ein Flußdiagramm der Schritte der Kompilierung von Fig. 15.



### Detaillierte Beschreibung des bevorzugten Ausführungsbeispiels

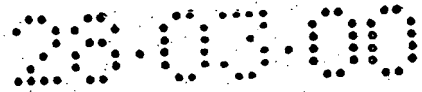
Ein in Fig. 1 dargestelltes hochverteiltes Industriesteuerungssystem ("HDIC") 10 enthält eine Anzahl von Steuerungsmodulen 12, die an ein gemeinsames Kommunikationsmittel 14 angeschlossen sind. Das Kommunikationsmittel 14 verwendet ein Kabel 16 aus Kupferleitern, längs denen digitale Nachrichten übertragen werden können, und, als Option, aus zwei Versorgungsleitern 18, die Steuerungsmodule 12 mit einer Gleichspannung von 24 Volt versorgen. Wie es im einzelnen weiter unten noch beschrieben wird, ist jedes Steuerungsmodul 12 parallel an das verdrehte Paar 16 angeschlossen, so daß zwischen irgendeinem Modul und allen anderen der Steuerungsmodule 12 eine Kommunikation stattfinden kann.

An das verdrehte Paar 16 des Kommunikationsmittels 14 ist auch ein Programmiergerät 20 angeschlossen, das einen Displayschirm 22, einen Prozessor 24 und ein Tastenfeld 26 hat. Bei dem Programmiergerät kann es sich um einen Allgemeinzweck-Personalcomputer handeln, wie er im Stand der Technik gut bekannt ist, wobei der Displayschirm 22 für die Ausgabe von Text und grafischen Darstellungen vom Prozessor 24 Sorge trägt und das Tastenfeld 26 die Eingabe alphanumerischer Zeichen in den Prozessor 24 gestattet. Das Programmiergerät 20 enthält auch ein Diskettenlaufwerk 30, das es gestattet, von auswechselbaren Magnetaufzeichnungsplatten (nicht gezeigt), die in das Laufwerk 30 eingeschoben werden können, Programme in das Programmiergerät 20 zu laden und aufzubewahrende Daten auf die Magnetaufzeichnungsplatten zu schreiben. Ein Cursorsteuerungsgerät beispielsweise eine Maus 28 kann mit dem Programmiergerät 20 verbunden sein, um die Manipulation grafischer Ikonen zu unterstützen, wie es weiter unten noch beschrieben wird.

Das Programmiergerät 20 enthält eine Schnittstelle zu dem verdrehten Paar 16, wie es noch im einzelnen in Verbindung mit den Steuerungsmodulen 12 weiter unten beschrieben wird, und diese Schnittstelle gestattet die Kommunikation zu den Steuerungsmodulen 12 in einer Art und Weise, die der Kommunikation zwischen den Steuerungsmodulen 12 ähnlich ist.

### HDIC-Anwendung auf ein Förderband

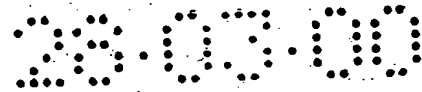
Unter Bezugnahme auf Fig. 1 und 2 enthält ein Ausführungsbeispiel eines hochverteilten Industriesteuerungssystems ("HDIC") 10 fünf Steuerungsmodule 12 zur Steuerung



eines einfachen Förderbandes 13. Das Förderband 13 enthält einen Förderriemen 32, der in Abhängigkeit vom Antrieb des Förderbandes durch einen Motor 16 auf seiner Oberseite Erzeugnisse 34 fortbewegen kann. Die Bewegung des Förderbandes wird durch Drücken einer beleuchteten Drucktaste 38 auf einem Drucktastenmodul 40 ausgelöst. Die Bewegung des Förderbandes 32 wird angehalten, sobald das Erzeugnis 34 auf einen Betätigungsarm 43 eines Grenzschaltermoduls 42 trifft oder wenn eine Stopdrucktaste 44 gedrückt wird. Die Stopdrucktaste ist ebenfalls in das Drucktastenmodul 40 einbezogen.

Es wird jetzt auch auf Fig. 3 bezug genommen. Das erste Steuerungsmodul 12, das in diesem HDIC-System 10 eingesetzt wird, ist ein Trennmodul 46, das eine Absicherung für die Drei-Phasen-Energie zum Motor 36 enthält und einen Trennschalter 64 aufweist, der durch einen Handgriff 48 betätigbar ist. Das Trennmodul 46 nimmt das Kommunikationsmittel 14 und auch eine Drei-Phasen-Versorgung auf, die über einen lösbaren Drei-Phasen-Bus 50 zugeführt wird und vom Modul an- oder abschaltbar ist. Im allgemeinen wird der Zustand des Ausgangs des Trennschalters, und zwar was das Anschalten oder Abschalten der Drei-Phasen-Energie vom Bus 50 anbelangt, über das Kommunikationsmittel 14 übertragen, wohingegen die tatsächliche Betätigung des Trennschalters manuell mittels des Handgriffs 48 ausgeführt wird.

Ein physischer Transformator 53, der die Gleichspannung von 24 Volt an die Energieleiter 18 liefert, ist typischerweise vor dem Trennmodul 46 angeschlossen, um die Netzwirkkommunikation selbst dann aufrecht zu erhalten, wenn das Trennmodul 46 die Drei-Phasen-Energie vom Steuerungssystem trennt. Bei diesem Beispiel ist der Transformator 53 kein Steuerungsmodul 12. Dennoch ist im Trennmodul ein virtueller Transformator 53' enthalten, der eine Primärwicklung hat, die an zwei Phasen der Drei-Phasen-Versorgung, die vom Trennschalter geschaltet wird, angeschlossen ist. Die Leitungen der Sekundärwicklung dieses virtuellen Transformators führen über einen virtuellen Leistungsschalter 55 zur Energiezuleitungsschiene 74 und über einen virtuellen Leistungsschalter 57 zur Energierückleitungsschiene 80 in der Steuerungslogik. Der virtuelle Transformator sieht eine Reihe von Systemeigenschaften vor, einschließlich: (1) das Öffnen des Trennschalters 64 schaltet die gesamte zugeordnete Steuerungslogik ab, (2) die virtuellen Leistungsschalter 55 und 57 sehen ein intuitives Verfahren vor, um ausgewählte Logik zu Editierzwecken abzuschalten.

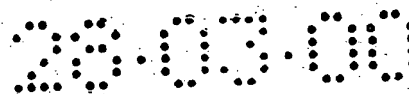


Wie es beschrieben wird, sind die Teile der Module 12 im allgemeinen virtuell über das Kommunikationsmittel 14 verbunden, also identisch zu den Verbindungen ihrer physischen Gegenstücke. Eine Ausnahme bildet der physische Transformator 53. Abweichend von dem virtuellen Transformator 53' ist der physische Transformator 53 vor dem Trennmodul 46 angeschlossen, so daß die Steuerungsmodule 12 ihre tatsächliche Versorgung nicht verlieren, wenn die Drei-Phasen-Versorgung vom Trennmodul 46 abgeschaltet wird.

Das zweite Steuerungsmodul 12 ist das Druckastenmodul 40, das, kurz beschrieben, drei Teile enthält: eine Startdrucktaste 38, eine Lampe 39 zum Beleuchten der Drucktaste 38, und eine nicht beleuchtete Stopdrucktaste 44. Das Druckastenmodul 40 sieht ein Steuerungsmodul 12 vor, das eine Benutzereingabe zum HDIC-System 10 gestattet.

Das dritte Steuerungsmodul 12 ist ein Grenzschaltermodul 42, das ein Beispiel für das einfachstmögliche Steuerungsmodul 12 ist und nur ein einziges Teil (einen Schalter) hat, der in Abhängigkeit von der Bewegung eines Betätigungsarms 43 nur eine einzige Schaltfunktion vorsieht.

Das vierte Steuerungsmodul 12 ist ein Motorstartmodul 52, dem, hinter dem Trennmodul 46 liegend, die Drei-Phasen-Energie vom Bus 50 zugeführt wird und das diese Energie an den Motor 36 weiterleitet. Das Motorstartmodul 52 enthält vier verschiedene Teile: 1) "Thermische" Überlastungselemente 70 für jede Phase, die den Stromfluß durch die Wicklungen des Motors 36 erfassen, um zu betätigen 2) normalerweise geschlossene Überlastungskontakte 78, und 3) eine Relaispule 72 zur Steuerung 4) normalerweise geöffneter Kontakte 68 in jeder Phase der Drei-Phasen-Energie, um bei Nichtvorhandensein einer Überlastungsbedingung den Motor 36 anzuschalten und abzuschalten, und 5) einen Sperrhilfskontakt 76. Der Überlastungsauslösepunkt der Überlastungselemente 70 wird durch das Programmiergerät 20 über das Kommunikationsmittel 14 eingestellt, wie es noch beschrieben wird. Es sei bemerkt, daß, obgleich bei diesem Beispiel die in Reihe miteinander verbundenen Kontakte 78 als ein einziges Teil betrachtet werden, jeder dieser Kontakte 78 auch unabhängig mit anderen Teilen verbunden und deshalb als separates Teil behandelt werden könnte. Um der Klarheit willen behandelt die folgende Diskussion prinzipiell die äußere Verdrahtung zwischen den Modulen anstelle der inneren Verdrahtung innerhalb der Module.



### Module, Teile und virtuelle Verbindungen

Im allgemeinen kann jedes Steuerungsmodul 12 irgendeine einer Vielzahl unterschiedlicher Funktionen haben und vorzugsweise grundsätzliche elektronische Komponenten nachbilden, wie sie in Industriesteuerungen verwendet werden. Die elementaren elektrischen Komponenten, die alleine oder in Kombination die Funktion der Steuerungsmodul 12 definieren, werden "Teile" genannt und sind im allgemeinen dadurch gekennzeichnet, daß sie zwei oder mehrere Anschlüsse haben, zwischen denen in Abhängigkeit vom Zustand des Teils ein Strom fließen kann. Somit enthält, wie bereits oben erwähnt, das Trennmodul 46 eine Anzahl Teile (Sicherungen und Kontakte). Im Gegensatz dazu enthält das Grenzschaltermodul 42 ein einziges Teil, das ein einpoliger Wechselkontaktschalter ist. Für einige Teile sind repräsentative Symbole in Fig. 5 zusammengestellt, die grafische Symbole gemäß einer Norm JIC aus einem JIC-Handbuch zeigt.

In noch größerer Allgemeinheit können die Anschlüsse Informationsflußbahnen darstellen, wie sie auftreten können in einem Vergleicher, einem monostabilen Kippglied, einem Addierglied oder einem Zähler, also in der digitalen Elektronik eingesetzten Bauelementen. Idealerweise verkörpert jedes Teil eine elementare Funktion, die im HDIC-System benötigt wird. Dies gestattet maximale Flexibilität bei der Verbindung der Komponenten des HDIC-Systems 10 und, wie es noch weiter unten diskutiert wird, sieht ein intuitives Verfahren zum Zuteilen des gesamten Steuerprogramms unter den Steuerungsmodulen 12 vor und gestattet eine maximale Auflösung für die Zwecke der Fehlersuche im HDIC-System 10.

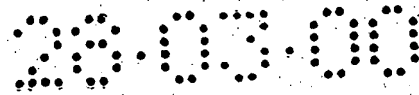
Wiederum unter Bezugnahme auf Fig. 1 sei erläutert, daß im allgemeinen alle Steuerungsmodul 12 über das Kommunikationsmittel 14 miteinander verbunden sind. Diese Art der Verbindung wird "Netzwerk"-Verbindung genannt. Bei der Implementation des Steuerprogramms können allerdings gewisse Teile jedes Steuerungsmoduls 12 nur auf ausgewählte andere Teile dieses oder anderer Steuerungsmodul 12 an dem Kommunikationsmittel 14 ansprechen. Die Teile der Steuerungsmodul kann man daher in einer solchen Weise betrachten, daß sie nur mit einigen der anderen Teile "virtuell" verbunden sind. Die Bahnen der virtuellen Verbindungen werden in einem Prozeß festgelegt, bei welchem Nachrichtenidentifizierungen von Nachrichten an dem Kommunikationsmittel 14 mit einer Verbindungsliste von Nachrichtenidentifizierungen verglichen werden, was im einzelnen weiter unten noch beschrieben wird.



Im Gegensatz zu Netzwerkverbindungen des Kommunikationsmittels 14 und den virtuellen Verbindungen zwischen Teilen haben gewisse Steuerungsmodule 12 auch "physische" Verbindungen zur Energieversorgung und zum gesteuerten Prozeß. So ist beispielsweise das Trennmodul mit der Drei-Phasen-Versorgungsenergie und dem Versorgungsbus 50 physisch verbunden, so daß dem Motor über das Motorstartmodul 52 tatsächlich eine Spannung zugeführt werden kann. Diese physische Verbindung, über die ein tatsächlicher Strom fließt, ist von der virtuellen Verbindung zu unterscheiden.

Unter Bezugnahme auf Fig. 3 können die gewünschten Bahnen der virtuellen Verbindungen zwischen Teilen in einer schematischen Darstellung repräsentiert werden, die jedes Teil in Form eines standardisierten oder genormten elektrischen Symbols darstellt. Die virtuellen Verbindungen zwischen Teilen sind durch durchgehende Linien 62 wiedergegeben, die zu den Anschlüssen an den Symbolen führen, die jedem Teil zugeordnet sind, wobei die Linien virtuelle Verbindungen zwischen Anhäufungen von Teilen innerhalb der Module 12 repräsentieren. Die als Rechtecke dargestellten Modulanschlüsse 60 hängen ab von der zufälligen Aggregation von Teilen innerhalb der Module 12 und erscheinen auf der schematischen Darstellung unter Generierung vom Programmgerät 20 in einem spezifischen Modus, und zwar nur dann, wenn die tatsächliche Modulhardware ausgewählt ist und die Aggregation bekannt ist, wie es weiter unten noch beschrieben wird.

So enthält beispielsweise das Trennmodul 46 sieben Teile: drei Kontakte 64 eines parallel betätigten Trennschalters, und zwar gesteuert vom Handgriff 48, drei in Reihe mit den Kontakten 64 geschaltete Sicherungen 66, und den virtuellen Transformator 53'. Die Teile sind innerhalb des Moduls 46 intern miteinander verbunden, so daß nur vier Modulanschlüsse 60 zur virtuellen Verbindung mit anderen Modulen vorhanden sind: einer für die stromabwärts gelegene Seite jeder Reihenschaltung aus Kontakt 64 und Sicherung 66 und einer für den Leiter des virtuellen Transformators zur Zufuhr von Energie zur Energiezuleitungsschiene 74. Für die stromaufwärts gelegene Seite jeder Reihenschaltung aus einem Kontakt 64 und einer Sicherung 66 sind keine Modulanschlüsse 60 gezeigt, weil stromaufwärts des Trennmoduls 46 kein Modul 12 angeschlossen wird. Weiterhin sind keine Modulanschlüsse 60 zwischen den Sicherungen 66 und den Kontakten 64 vorgesehen, weil dort kein Modul 12 angeschlossen wird.



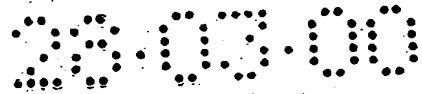
Jeder Modulanschluß 60 ist entweder ein stromaufwärts liegender Anschluß, der von der Energiezuleitungsschiene 74 direkt oder über andere Module 12 Strom empfängt, oder ein stromabwärts liegender Anschluß, der der Energierückleitungsschiene 80 Strom direkt oder über andere Module 12 zuführt. Mit einer Ausnahme ist der am weitesten links vorgesehene Anschluß der stromaufwärts liegende Anschluß 60 und der am weitesten rechts vorgesehene Anschluß der stromabwärts liegende Anschluß 60, und zwar unter Beachtung der Konvention, daß die Energie von links nach rechts fließt. Die Ausnahme bildet der virtuelle Transformator 53, wo der stromabwärts liegende Anschluß 60 zur linken vorgesehen ist, um im Schemabild Platz zu haben.

Die Gruppierung der Teile in Fig. 3 innerhalb der gestrichelten Linien gibt die physische Gruppierung innerhalb der Module 12 von Fig. 1 an. Diese physische Gruppierung wird erlaubterweise für die Zwecke der Konstruktion des Schemabildes ignoriert, bei dem idellerweise die virtuellen Verbindungen der Teile ihre jeweilige Platzierung im Schemabild diktieren. Das grafische Symbol irgendeines Teils eines Moduls 12 kann sich irgendwo in der schematischen Darstellung befinden.

In den tatsächlichen elektrischen Vorrichtungen, die von den Teilen der Steuerungsmodule 12 nachgebildet werden, kann der Strom ohne Beschränkung in jeder der beiden Richtungen zwischen den Anschlüssen fließen, wie es für einen Schalter der Fall ist. Wie bereits oben erwähnt, haben allerdings die virtuellen Verbindungen zwischen den Teilen der Steuerungsmodule 12 durch Vorgabe eine eingeschränkere Konvention zu beachten, daß nämlich der Strom nur in einer Richtung zwischen den Anschlüssen der Teile fließt. Es gibt daher einen bestimmten stromaufwärts liegenden oder Eingangsanschluß und einen bestimmten stromabwärts liegenden oder Ausgangsanschluß, wie es durch die Richtung des Stromflusses definiert ist. Diese Konvention spiegelt sich wider in den Symbolen, die verwendet werden, um die Teile zu repräsentieren, wobei ein Stromfluß nur von dem am weitesten links liegenden Anschluß des Symbols zu dem am weitesten rechts liegenden Anschluß erfolgen kann. Wie es aus der nachstehenden Beschreibung hervorgeht, kann man allerdings die Konvention eines bidirektionalen Stromflusses durch das bei der Erfindung benutzte System virtueller Verbindungen leicht implementieren.

Unter Bezugnahme auf Fig. 1 und 3 sind die Teile der Steuerungsmodule 12 virtuell und physisch wie folgt verbunden: Das Trennmodul 46 erhält dreiphasige Energie über eine

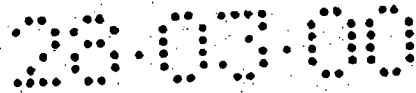




physische Verbindung zu der Drei-Phasen-Energieversorgung - die virtuelle Verbindung ist impliziert. Auf der stromabwärts liegenden Seite ist der Trennschalter 46 physisch mit dem Drei-Phasen-Bus 50 verbunden, um Energie an den Motorstarter 52 und damit an den Motor 36 zu liefern. In entsprechender Weise sind stromabwärts liegende Anschlüsse 60 des Trennmoduls 46 virtuell über Leitungen 62 mit dem stromaufwärts liegenden Anschluß 60 des Motorstartmoduls 52 verbunden, und zwar unter Spiegelung dieser physischen Verbindung. Es gibt eine interne virtuelle Verbindung zwischen der geschalteten Drei-Phasen-Energie und der Primärwicklung des virtuellen Transformators 53'. Ein stromabwärts liegender Anschluß 60 der Sekundärwicklung des virtuellen Transformators 53' ist über die Energiezuleitungsschiene 74 mit den anderen stromaufwärts liegenden Anschlüssen verbunden.

Der stromaufwärts liegende Anschluß 60 der Drucktaste 38 im Modul 40 ist ebenfalls virtuell mit dem virtuellen Transformator 53' über die Energiezuleitungsschiene 74 verbunden, wobei es allerdings keine physische Verbindung gibt. Der stromabwärts liegende Anschluß 60 der Drucktaste 38 ist wiederum virtuell mit dem Eingangsanschluß der Drucktaste 44 verbunden. Dieser Eingangsanschluß der Drucktaste 44 ist auch virtuell an den Ausgangsanschluß 60 eines Sperrhilfskontakts 76 angeschlossen, der ebenfalls virtuell mit der Energiezuleitungsschiene 74 verbunden ist, um die Anschlüsse der Drucktaste 38 kurzzuschließen und dadurch eine Sperrwirkung vorzusehen, wie es allgemein üblich ist. Der Sperrhilfskontakt 76 ist ein normalerweise geöffneter Hilfskontakt, der auch von der Spule 72 angesteuert wird. Keine dieser virtuellen Verbindungen hat wiederum entsprechende physische Verbindungen.

Der Ausgang der Drucktaste 44 ist virtuell mit dem stromaufwärts liegenden Anschluß 60 des Pols des Grenzschaltermoduls 42 verbunden, dessen normalerweise geschlossener Kontakt über einen Ausgangsanschluß 60 mit dem Eingangsanschluß 60 des Moduls 52 verbunden ist, der zu einer Reihenschaltung aus drei normalerweise geschlossenen Kontakten führt, die durch das thermische Überlastungselement 70 gesteuert werden, um bei einer thermischen Überlastung zu öffnen. Diese normalerweise geschlossenen Kontakte 78 sehen eine virtuelle Bahn zu einem Anschluß der Spule 72 vor, und dann durch die Spule 72 und einen weiteren Anschluß 60 zu einer Energierückleitungsschiene 80. Eine Lampe 39, die zur Beleuchtung der Drucktaste 38 dient, ist virtuell zwischen der Energierückleitungsschiene 80 und dem Eingangsanschluß der Spule 72 geschaltet, um eine Anzeige darüber zu liefern, daß die Spule 72 erregt worden ist.



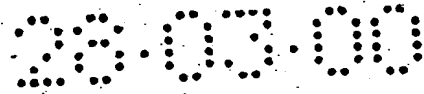
Die obige Beschreibung ist im wesentlichen identisch mit den physischen Drahtverbindungen von tatsächlichen Teilen, wie Spulen, Drucktasten und Grenzscharter, wie sie zum implementieren des oben beschriebenen Fördersystems benutzt werden. Bei den Verbindungen handelt es sich allerdings nicht um tatsächliche Drahtverbindungen, sondern einfach um Kommunikationsbahnen, die durch Nachrichten auf dem Kommunikationsmittel 14 erwirkt werden, wie es weiter unten im einzelnen noch beschrieben wird.

### Modul-Hardware

Unter Bezugnahme auf Fig. 4 hat jeder Steuerungsmodul 12 Anteil an einer Anzahl von Eigenschaften. Jeder Steuerungsmodul 12 kann die Versorgungs- oder Energieleiter 18 über eine Energiekonditionierschaltung 105 aufnehmen, die einen Schutz gegen umgekehrte Polarität und Hochspannungsspitzen an den Energieleitern 18 vorsieht, wie es auf dem Gebiet der industriellen Elektronik bekannt ist.

Wie es auch oben beschrieben ist, empfängt jeder der Steuerungsmodule 12 das verdrehte Leiterpaar 16 des Kommunikationsmittels 14. Speziell wird das verdrehte Paar 16 von zwei Anschlüssen eines auswechselbaren Anschlußstreifens 84 empfangen und ist mittels Brücken in einer verketteten Art und Weise ("daisy chain") angeschlossen. Das Signal am verdrehten Paar 16 wird vom Anschlußstreifen 84 bei den ersten beiden Anschlüssen abgenommen und gelangt von dort zu einem Sendeempfänger 85 und dann zu einer CAN-Verbindungssteuerung 86, die die notwendige Zeitgabe und Bitsequenz für das Aussenden und Empfangen von Nachrichten längs des Kommunikationsmittels 14 vorsieht.

Die CAN-Verbindungssteuerung 86 sieht für den Empfang und das Aussenden formatierter Nachrichten längs des Kommunikationsmittels 14 eine speicherabgebildete Schnittstelle vor. Ein Beispiel einer CAN-Verbindungssteuerung 86 ist eine 80C592-Mikrosteuerung, die zum Ausführen der CAN-Protokolle vorprogrammiert ist und im Handel erhältlich ist, beispielsweise von Signetics Corporation, Sunnyvale, Kalifornien. Die CAN-Spezifikation definiert die Anforderungen, die die Mittelzugriffssteuerung (MAC = Media Access Control) und physischen Signalisierschichten der ISO/OSI für Datenkommunikationsnetzwerke umfassen. Die CAN-Protokolle sind spezifiziert in einem ISO-Dokument ISO/TC22/SC3/WG1, autorisiert durch Robert Bosch GmbH und hiermit unter Bezugnahme einbezogen. Die CAN-



Norm gestattet im allgemeinen die Kommunikation asynchroner digitaler Nachrichten längs eines verdrehten Leiterpaares, wie es im Kommunikationsmittel 14 enthalten ist, mit Geschwindigkeiten bis zu 1 Mbit/Sekunde. Das CAN-Protokoll spezifiziert eine Null- bis Acht-Byte-Datennachricht einschließlich eines Elf-Bit-Arbitrationsfeldes, das die Priorität festlegt, falls zwei Knoten Nachrichten zur selben Zeit senden wollen.

Die CAN-Verbindungssteuerung 86 enthält einen internen Pufferspeicher (nicht gezeigt), der direkt adressiert werden kann. Jeder Platz des Puffers kann in ein Arbitrationsfeld geladen werden, und zwar zur Verwendung beim Aussenden und Empfangen von Nachrichten. Im Falle des Aussendens von Nachrichten wird der Nachrichtenpuffer auch mit Nachrichtendaten geladen. Die CAN-Verbindungssteuerung 86 sendet automatisch die Nachrichten aus, wenn das Kommunikationsmittel 14 gemäß der Priorität seines Arbitrationsfeldes frei ist, und signalisiert dann, daß die Nachricht ausgesendet worden ist.

Im Falle des Empfangs von Nachrichten wird eine ankommende Nachricht mit den Arbitrationsfeldern jedes Pufferplatzes verglichen, und der zuerst aufgefundene Platz mit dem selben Arbitrationsfeld erhält die Nachricht. Zur Zeit des Empfangs wird ein Signal erzeugt, das den Empfang anzeigt.

Die CAN-Verbindungssteuerung 86 kommuniziert über einen Bus 92 mit einem Festwertspeicher ROM 94, einem flüchtigen RAM 96, einem nichtflüchtigen RAM 98, einem Mikroprozessor 100 und einer digitalen E/A-Schaltung 102, wie es unter Bezugnahme auf den Stand der Technik verständlich ist und wie es innerhalb eines einzigen zugeordneten Mikroprozessors 90 implementiert werden kann.

Die digitale E/A-Schaltung 102 kommuniziert in Abhängigkeit von den Eigenschaften des Steuerungsmoduls 12 mit verschiedenen modulspezifischen Eingängen und Ausgängen. Für das in Fig. 4 dargestellte Druckastenmodul 40 liefert die digitale E/A-Schaltung 102 beispielsweise ein Signal zur Beleuchtung der Druckastenlampe 39, und sie erhält ein Signal das anzeigt, ob die Druckaste 38 gedrückt worden ist. Das Druckastenmodul 40 enthält ein physisches Betätigungsglied, nämlich den Schaft 108 der Druckaste, der aus dem Gehäuse des Steuerungsmoduls 12 herausragt und gegen die Kraft einer Feder 110 nach innen gedrückt werden kann. Das innere Ende des Schaftes 108 arbeitet mit einem Magneten 112 zusammen, mit dessen Hilfe ein elektrisches Signal erzeugt wird, wenn der Schaft 108 längs eines Hall-

Effekt-Detektors 114 verschoben wird. Dieses Signal wird von der E/A-Schaltung 102 empfangen.

Die verschiedenen physischen Komponenten, die im Steuerungsmodul 12 verkörpert sind, ändern sich in Abhängigkeit von den Eigenschaften des Steuerungsmoduls. So ist beispielsweise beim Grenzschalterm modul 42 die Lampe 39 nicht vorhanden, und der verschiebbare Schaft 108 kann eine Radialbewegung ausführen, damit er dem Betätigungsarm 43 des Grenzs Schalters angepaßt ist. Gleichermaßen kann der Motorstarter 52 eine Schaltung enthalten, die zum Anzeigen von Stromfluß zum Motor 36 und zum Bereitstellen eines Signals dient, das eine Öffnung des Kontakts 78 bewirkt, wenn dieser Strom einen vorbestimmten Wert überschreitet. Diese physischen Signale, die externe Aspekte des HDIC-Systems 10 betreffen, werden über die digitale oder analoge E/A-Schaltung 102 vom Bus 92 empfangen.

Jedes Steuerungsmodul 12 kann auch eine oder mehrere Status-Anzeige-LEDs 106 enthalten, die im allgemeinen anzeigen, daß das Steuerungsmodul 12 gerade mit Energie versorgt wird und das ein im ROM 94 abgelegtes Programm vom Mikroprozessor 100 gerade erfolgreich ausgeführt wird, wie es auf dem Gebiet dieser Technik üblich ist.

Jedes Steuerungsmodul 12 kann auch einen externen zugänglichen Schalter 116 aufweisen, der sechs Bits digitaler Information zu den E/A-Schaltungen 102 übertragen kann. Dieser Schalter 116 gestattet es, daß jedem Steuerungsmodul 12 eine eindeutige Knotenadresse zugeordnet werden kann, die dazu dient, Nachrichten zu identifizieren, welche von diesem Modul an das Kommunikationsmittel 14 ausgesendet werden, wie es noch beschrieben wird.

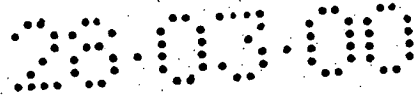
Es sei zusammengefaßt, daß somit der Mikroprozessor 100 ein gespeichertes Programm vom ROM 94 empfangen, Nachrichten vom Kommunikationsmittel 14 empfangen und an es aussenden, den Zustand verschiedenartiger physischer Komponenten, die diesem Modul zugeordnet sind, wie beispielsweise den Drucktastenschaft 108, die Lampe 39 und die LEDs 106, verändern oder erfassen, und gewisse Variable, die während der Ausführung des im ROM 94 abgelegten Programms benutzt werden, im flüchtigen RAM 96 oder im nicht-flüchtigen RAM 98 speichern oder daraus auslesen kann.

### Datenstrukturen im Modulspeicher

Es wird jetzt auf Fig. 3 und 7 bezug genommen. Ein Abschnitt des nichtflüchtigen RAM 98 speichert eine Stromaufwärtsverbindungsliste 118, die im allgemeinen besteht aus einem Satz Nachrichtenidentifizierungen von Steuerungsmodulen 12 und Teileanschlüssen 60' spezifischer Teile in diesen Steuerungsmodulen 12, die (1) virtuell verbunden sind mit jedem gegebenen Anschluß 60 des Steuerungsmoduls 12, und zwar (2) unmittelbar stromaufwärts des gegebenen Steuerungsmoduls 12. Teileanschlüsse 60' können Modulanschlüsse 60 sein, wie es oben beschrieben worden ist, oder Anschlüsse von Teilen, die keine Verbindungen zur Außenseite des Moduls 12 haben.

Die Nachrichtenidentifizierungen können eine Kombination aus der Knotenadresse des Moduls 12, das den Teileanschluß 60' enthält, und aus einer intern definierten Anschlußadresse sein, was von der Funktionalität des Moduls 12 und seiner Anzahl von Teileanschlüssen abhängt. Die Nachrichtenidentifizierung gestattet es einem gegebenen Teileanschluß 60' in einem gegebenen Steuerungsmodul 12 eindeutig bezeichnet zu werden. Stromaufwärtsverbindungen sind virtuelle Verbindungen, die direkt vorangehen, oder sie haben eine Bahn durch Drähte 62 und andere Steuerungsmodule 12, und zwar stromaufwärts der Energiezuleitungsschiene 74 unter Beachtung der Konventionen für den Stromfluß für die Teile. Unmittelbare Stromaufwärtsverbindungen sind solche, die sowohl stromaufwärts sind, aber direkt durch Drähte 62 mit dem gegebenen Teileanschluß 60 verbunden sind, ohne durch andere Steuerungsmodule 12 zu laufen. Für ein Steuerungsmodul 12, der einen einfachen Schalter nachbildet, beispielsweise ein Grenzschaltermodul 42, ist die Stromaufwärtsseite diejenige Seite, die in der physischen Realisation des Schalters Stromfluß empfangen würde.

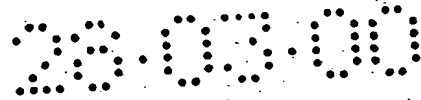
Die Stromaufwärtsverbindungsliste 118 beschreibt somit jede unmittelbare Stromaufwärtsverbindung für jeden Teileanschluß 60' des gegebenen Steuerungsmoduls 12. Ist ein Teileanschluß 60' direkt mit der Energiezuleitungsschiene 74 verbunden, wird die Nachrichtenidentifizierung des Leistungsschalters 55 (01-3) in die Stromaufwärtsverbindungsliste 118 eingegeben. Hat der Teileanschluß 60' keine Verbindungen, wird keine Nachrichtenidentifizierung in die Stromaufwärtsverbindungsliste 118 eingetragen. Wie es unten beschrieben wird, spricht jedes Steuerungsmodul 12 im allgemeinen nur auf Nachrichten am Kommunikationsmittel 14 an, die ihren Ursprung von Steuerungsmodulen 12 in seiner Stromaufwärtsverbindungsliste 118 haben.



Eine als Option vorgesehene Stromabwärtsverbindungsliste 120, die sich auch im nichtflüchtigen RAM 98 befindet, und zwar in einer zur Stromaufwärtsverbindungsliste 118 analogen Weise, kann einen Satz Nachrichtenidentifizierungen enthalten, die Anschlüsse 60 bezeichnen, welche virtuell verbunden sind mit jedem Teileanschluß 60' des gegebenen Steuerungsmoduls 12 und welche unmittelbare Stromabwärtsnachbarn zu dem gegebenen Modul 12 sind. Stromabwärtsnachbarn sind solche, die längs einer Bahn virtueller Verbindungen letztlich zur Energierückleitungsschiene 80 führen. Ist ein Teileanschluß 60' direkt mit der Energierückleitungsschiene 80 verbunden, wird die Nachrichtenidentifizierung des Leistungsschalters 57 (01-3) in die Stromabwärtsverbindungsliste 120 eingetragen. Hat der Teileanschluß 60' keine Stromabwärtsverbindungen, wird keine Nachrichtenidentifizierung in die Stromabwärtsverbindungsliste 120 eingetragen. Das gegebene Steuerungsmodul ist reichlich indifferent zu seinen unmittelbaren Stromabwärtsnachbarn, ausgenommen in einem Reparaturmodus, was später noch beschrieben wird.

Es wird jetzt auf Fig. 4, 6 und 7 bezug genommen. Die Stromaufwärtsverbindungsliste 118 und die Stromabwärtsverbindungsliste 120 dienen dazu, jedes der Teile jedes Steuerungsmoduls 12 mit den anderen Teilen der anderen Steuerungsmodule 12 unabhängig von den tatsächlichen Netzwerksverbindungen zu verbinden, und zwar in einer Weise, die einfach durch Umprogrammierung der Stromaufwärtsverbindungsliste 118 und der Stromabwärtsverbindungsliste 120 geändert werden kann.

Unter Bezugnahme auf Fig. 3 und 6 sowie den oben erläuterten Umstand, daß die Nachrichtenidentifizierungen jeden Teileanschluß 60' eindeutig identifizieren, hat jedes Steuerungsmodul 12 eine Moduladresse und jeder Teileanschluß 60' eines Teils innerhalb des Steuerungsmoduls 12 hat eine Anschlußadresse. Im Beispiel von Fig. 3 kann das Trennmodul 46 eine Modulnetzwerkadresse von 01, das Druckastenmodul 40 eine Modulnetzwerkadresse von 02, das Grenzschalterm modul 42 eine Modulnetzwerkadresse von 03 und das Motorstartmodul 52 eine Modulnetzwerkadresse von 04 haben. Gleichmaßen können die Anschlüsse 60 innerhalb jedes Steuerungsmoduls 12 sequentiell per Stück durchnummeriert sein. Bei einer bevorzugten Ausführungsform sind die Eingangsanschlüsse sequentiell von Null bis zur Gesamtzahl der Eingangsanschlüsse und in ähnlicher Weise die Ausgangsanschlüsse sequentiell von Null bis zur Gesamtzahl der Ausgangsanschlüsse durchnummeriert. Die Eingangsanschlüsse des Trennmoduls 46 können beispielsweise die Anschlußadressen 0, 1 und 2 und die



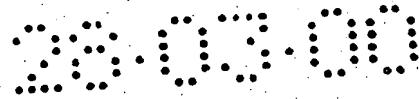
Ausgangsanschlüsse, für entsprechende Teile des Trennmoduls 46 können die Anschlußadressen 0, 1 und 2 haben.

Die Eingangsanschlüsse 60 des Motorstartmoduls 52, die zu den Kontakten 68 führen, können virtuell mit den Ausgangsanschlüssen 60 des Trennmoduls 46 dadurch verbunden werden, daß die Nachrichtenidentifizierungen dieser Ausgangsanschlüsse 60 in die Stromaufwärtsverbindungsliste 118 des Motorstartmoduls 52 eingetragen werden. Die Verbindungsliste 118, die dem Eingangsanschluß für einen Eingang des Motorstartmoduls 52 zugeordnet ist, hat somit die Nachrichtenidentifizierung 01-0, die anzeigt, daß er mit dem Steuerungsmodul 12 mit der Moduladresse 01 und mit dem Teileanschluß 60' mit der Anschlußadresse 0 verbunden ist. Die derart gespeicherte Nachrichtenidentifizierung erzeugt eine virtuelle Verbindung 62 zwischen diesen beiden Anschlüssen 60.

Bei der bevorzugten Ausführungsform ist die virtuelle Verbindung bidirektional. Das bedeutet, daß die Stromabwärtsverbindungsliste 120 für das Trennmodul 46 auch die Nachrichtenidentifizierungen seines Stromaufwärtsnachbars in dem Motorstartmodul 52 enthält. Die Stromabwärtsverbindungsliste 120 für den Ausgangsschluß 0 des Trennmoduls 46 enthält somit die Adresse 04-3, die anzeigt, daß er verbunden ist mit dem Eingangsanschluß 3 des Moduls der Adresse 04 (dem Motorstarter). Dennoch wird für den Routinebetrieb des HDIC-Systems 10 lediglich der Aufwärtsverbindungsspeicher benötigt.

Ein gegebener Teileanschluß 60' kann virtuell verbunden werden mit mehr als einem anderen Anschluß durch Mehrfacheintragen in die Stromaufwärtsverbindungsliste 118. So ist beispielsweise der Eingangsanschluß 1 des Drucktastenmoduls 40, der den Stromaufwärtsanschluß der Drucktaste 44 darstellt, sowohl mit dem Ausgangsanschluß 0 der Modulnummer 02, und hat daher einen Eintrag 02-0 in seiner Aufwärtsverbindungsliste 118, als auch mit dem Ausgangsanschluß 2 des Moduls 04 (der Motorstarter) verbunden, und hat somit einen Eintrag von 04-2 in seiner Stromaufwärtsverbindungsliste 118. Wie man sieht, erzeugen Mehrfacheintragen in der Stromaufwärtsverbindungsliste 118 für einen gegebenen Anschluß ein logisches "verdrahtetes Oder", was bedeutet, daß der Eingangsanschluß auf Signale von jedem der beiden Stromaufwärtsnachbarn anspricht.

In ähnlicher Weise kann der Stromabwärts- oder Ausgangsteileanschluß 60' irgendeines Teils mit mehrfachen Stromaufwärts- oder Eingangsanschlüssen nachfolgender Steue-

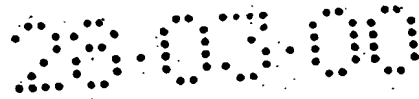


rungsmodule 12 verbunden werden. So ist beispielsweise der Ausgangsanschluß 60' des Satzes normalerweise geschlossener Kontakte 78, die dem Motorstartmodul 52 zugeordnet sind, sowohl mit dem Ausgangsanschluß 1 des Motorstartmoduls 52 als auch dem Eingangsanschluß der Spule 72 und auch mit dem Eingangsanschluß 2 des Drucktastenmoduls 40 verbunden, bei dem es sich um den Stromaufwärtsanschluß der Lampe 39 handelt. Diese Verbindung bedeutet, daß die beiden Steuerungsmodule 40 und 54 von Signalen Gebrauch machen, die vom Ausgangsanschluß 0 des Motorstartmoduls 52 stammen. Die Verbindung des Anschlusses 0 des Motorstartmoduls 52 mit dem Anschluß 1 des Motorstartmoduls 52 muß nicht die Erzeugung einer Nachricht auf dem Kommunikationsmittel 14 zur Folge haben, sondern kann intern vom Mikroprozessor 100 gehandhabt werden, wie es beschrieben wird. Dennoch liegt eine vollständige Flexibilität vor, irgendeinen Stromabwärtsteileanschluß 60' mit irgendeinem anderen Stromaufwärtsteileanschluß 60' desselben oder eines anderen Steuerungsmoduls 12 zu verbinden.

Das betrachtete HDIC-System 10 gestattet es, daß auf der Grundlage der Funktionalität der Teile eines Steuerungsmoduls 12 den Verbindungen zwischen den Steuerungsmodulen 12 gewisse Regeln auferlegt werden können. Die Regeln können beispielsweise die direkte Verbindung der Energiezuleitungsschiene 74 mit der Energierückleitungsschiene 80 verbieten, da sonst ein virtueller Kurzschluß vorliegen würde. Diese Restriktionen werden allerdings durch das oben beschriebene Zwischenverbindungssystem, das von den Stromaufwärts- und Stromabwärtsverbindungslisten 118 und 120 Gebrauch macht, nicht gefordert. In der Tat können solche ungültigen Verbindungen benutzt werden, um die Arbeitsweise der virtuellen Leistungsschalter 55 und 57 zu überprüfen.

Unter Bezugnahme auf Fig. 7 sei erläutert, daß der nichtflüchtige RAM 98 auch eine Teilelogiktablette 122 enthält, die eine begrenzte Modifikation der Logik der Teile des Steuerungsmoduls 12 gestattet. Die grundsätzliche Teilelogik ist im allgemeinen im ROM 94 enthalten, und zwar in Teilelogikdaten 104, die die Teilelogik definieren. So besteht beispielsweise die Teilelogik für eine Drucktaste 38 darin, daß, wenn sich die Drucktaste in einem "freigegebenen" Zustand befindet, der Ausgangsanschluß eines normalerweise geöffneten Kontaktes "niedrig" ist, und, wenn sich die Drucktaste in einem "gedrückten" Zustand befindet, der Ausgangsanschluß einen Wert hat, der gleich dem Wert des Eingangsanschlusses ist. Diese fundamentale Logik kann durch die Teilelogiktablette 122 beeinträchtigt werden, daß man die Interpretation der Position des Operators 108 in Bezug darauf ändert, ob der Schalter

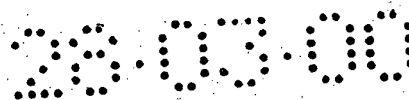




normalerweise geöffnet oder normalerweise geschlossen ist. Dies bedeutet, daß der Schalter so programmiert werden kann, daß er entweder normalerweise geschlossen oder normalerweise geöffnet ist. Die Veränderungsmöglichkeit der Teilelogik hängt von den Teilen ab, die im Steuerungsmodul 12 enthalten sind. Betrachtet man beispielsweise einen Drehschalter, dann können die Varianten des normalerweise geöffneten und normalerweise geschlossenen Zustands durch die Varianten Schließen-Vor-Öffnen oder Öffnen-Vor-Schließen ersetzt werden. Bei einem Schließen-Vor-Öffnen-Schalter tritt ein Kurzschluß zwischen benachbarten Stellungen während des Schaltvorgangs auf, wohingegen bei einem Öffnen-Vor-Schließen-Schalter benachbarte Stellungen während des Schaltvorgangs nicht kurzgeschlossen werden. Die Teilelogiktafel 122 kann auch Schwellenwertgrößen enthalten, beispielsweise den thermischen Überlastungsstrom, bei dem sich das thermische Überlastungselement 70 im Motorstartmodul 52 öffnet.

Der nichtflüchtige RAM 96 kann auch Programmierkennungen speichern, die von einem Programmierer verwendet werden, um während der Programmierung des HDIC-Systems 10 sowohl auf die Teile als auch die virtuellen Verbindungen Bezug zu nehmen, wie es im einzelnen weiter unten noch beschrieben wird. Eine dieser Programmierungshilfen ist eine Verdrahtungsliste 124, die den virtuellen Verbindungen zwischen den gegebenen Steuerungsmodulen 12 "Drahtnummern" zuordnet, wenn die virtuellen Verbindungen in schematischer Form dargestellt werden. Diese Drahtnummern machen das letzte Programm des HDIC-Systems 10 für einen menschlichen Programmierer dadurch leichter zugänglich, daß die Komplikation vermieden wird, die virtuellen Verbindungen durch Anschlußadressen zu definieren, die gerätespezifisch sind. Diese gespeicherte Information gestattet es auch, daß das Programmiergerät 20 das Steuerprogramm direkt von den mit dem System verbundenen Modulen 12 rekonstruieren kann.

Eine Modulkennung 126, bei der es sich um eine einfache Textfolge handelt, die die Funktion des Steuerungsmoduls 12 angibt, kann auch im nichtflüchtigen RAM gespeichert werden. Für die Drucktaste 38 des betrachteten Beispiels kann man die allgemeine Bezeichnung "1LPB" für die erste beleuchtete Drucktaste verwenden. Sowohl die Nummern der Drahtliste als auch die Kennungen sind in Bezug auf die tatsächliche Funktion des Steuerungsmoduls 12 im wesentlichen willkürlich und werden dem Steuerungsmodul 12 während der Anfangsprogrammierung gegeben. Sie werden lediglich benötigt, falls es erforderlich ist, das Gesamt-HDIC-Programm in der ursprünglich eingegebenen Form aus seinen verteilten

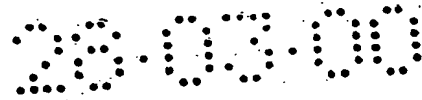


Abschnitten zu rekonstruieren, wie sie letzten Endes mit den verschiedenartigen Steuerungsmodulen 12 als Stromaufwärts- und Stromabwärtsverbindungslisten dargestellt sind.

Der ROM 94 enthält Information, die während der Programmierung des Steuerungsmoduls 12 nicht verändert wird und die eine Modul-ID-Nummer 128 umfaßt, welche die permanenten physischen Eigenschaften des Steuerungsmoduls identifiziert, wie beispielsweise, ob das Modul einen Drucktastenoperator 108 oder eine Lampe 39 hat. Die Modul-ID ermöglicht es auch, daß die Programmierungssoftware für das gesamte HDIC-System 10 das Steuerungsmodul 12 in Übereinstimmung mit den Programmierattributen des Moduls bringt: Wie beispielsweise seine physische Packungsikone und logische Überprüfungsregeln, als auch mit anderen Teileattributen, wie beispielsweise die Anzahl von Anschlüssen innerhalb des Steuerungsmoduls 12 und die Teilelogik jedes Teils. Die aktuellen Programmier- und Teileattribute können im Modul 12, im Programmiergerät 20 gespeichert sein, und zwar wie dort geladen durch Magnetplatten- oder andere gut bekannte Verfahren.

Alternativ und bei der bevorzugten Ausführungsform sind diese Attribute im ROM 94 abgelegt. Insbesondere sind Programmierattribute in einer Ikonengrafiktabelle 130 und in einer Logiküberprüfungsregeltabelle 132 innerhalb des ROM's 94 enthalten. Die Ikonengrafiktabelle 130 enthält ein Bitabbild eines Symbols, wie beispielsweise eines der in Fig. 5 dargestellten Symbole, wobei dieses Symbol auf die Funktion der Teile innerhalb des Steuerungsmoduls 12 bezogen ist. Vorzugsweise sind wenigstens zwei unterschiedliche Ikonen für jeden Zustand jedes Teils des Steuerungsmoduls 12 sowie eine Ikone, die das Steuerungsmodul 12 in seiner Gesamtheit darstellt, verfügbar. Diese zuletzt genannte Gesamtsteuerungsmodulikone ist "geöffnet", wie es weiter unten beschrieben wird, damit die Teileikonen zu sehen sind. Bezüglich der Teileikonen können beispielsweise sowohl geöffnete, geschlossene und gestörte symbolische Darstellungen eines Schalterteils in der Ikonengrafiktabelle 130 enthalten sein. Diese Abbildungen der Ikonengrafiktabelle 130 können hochgeladen werden in ein Programmiergerät 20, wie es weiter unten noch beschrieben wird.

Eine zweite Tabelle 132 mit Programmierattributen enthält Logiküberprüfungsregeln, bei denen es sich um Routinen handelt, die es dem Programmierer gestatten, ungeeignete Verbindungen der Teile des gegebenen Steuerungsmoduls 12 mit anderen Teilen zu entdecken. So sollte beispielsweise ein Spulenteil, wie die Spule 72, nicht durch eine virtuelle Verbindung von ihrem Eingangsanschluß zu ihrem Ausgangsanschluß kurzgeschlossen sein.



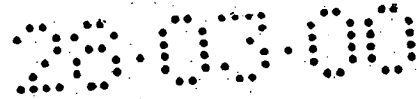
Wird eine solche Verbindung gemacht, werden sich die virtuellen Leistungsschalter 55 und 57 öffnen.

Die Teileattribute der Teile innerhalb eines Steuerungsmoduls 12 sind vorzugsweise ebenfalls im ROM 94 abgelegt. In der Hauptsache sind dies die Anschlußadressen der Teile, wie sie in einer Anschlußliste 134 enthalten sind. Die Anschlußliste 134 zeigt auf, wieviele Anschlüsse innerhalb des Steuerungsmoduls 12 vorgesehen sind, und sie zeigt ferner ihre Adressen an und, ob es sich um Eingangsanschlüsse oder Ausgangsanschlüsse handelt, und welche Logik ihnen zugeordnet ist. So würde beispielsweise für das Drucktastemodul 40 die Anschlußliste 134 darstellen, daß es sechs Instanzen von Anschlüssen gibt, und zwar mit der Bezugnahme 0 bis 5, wobei die Anschlüsse 0 ein Schalter, die Anschlüsse 1 ein Schalter und die Anschlüsse 2 eine Lampe sind. Dies gestattet es dem Programmiergerät 20, die geeignete Anzahl von Anschlüssen und die Orientierung der Anschlüsse für die Programmierung durch den Anwender darzustellen. Eng bezogen auf die Anschlußliste 134 ist die Teilelogiktafel 104, die das Programmiergerät mit einer Anzeige der Funktion der verschiedenartigen Teile durch ihre Logik ausstattet.

Trotz der Gruppierung der Teile innerhalb der Module 12 werden allerdings alle Teile, die innerhalb des Moduls 12 enthalten sind, dennoch für den Anwender als voneinander unabhängige Elemente in einer traditionellen "Materialstückliste" dargestellt, und dem Anwender ist es völlig freigestellt, ohne bezug auf die tatsächlichen Module 12, die die Teile enthalten, Teile miteinander zu verbinden.

### Programmierung des HDIC-Systems

Es wird jetzt auf Fig. 1, 7 und 8 bezug genommen. Die anfängliche Programmierung des HDIC-Systems kann durchgeführt werden entweder erstens "on-line" über das Programmiergerät 20, das mit dem Kommunikationsmittel 14 verbunden ist, oder zweitens "off-line" durch den Programmierer, der die Hardware bezeichnet, die in dem gewünschten Steuerungssystem mit dem Netzwerk verbunden ist. Im allgemeinen ist keine besondere Reihenfolge für die zu beschreibenden off-line-Programmierschritte vorgeschrieben, jedoch werden sie, aus Gründen der Einfachheit, unter Bezugnahme auf ein lineares Voranschreiten von einer Anfangskonfiguration bis hin zur Kompilierung und zum Herabladen dargestellt und erläutert.

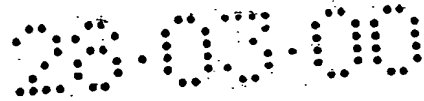


Der Programmierer benutzt im Programmiergerät 20 vorhandene Programmiersoftware, um das Steuerprogramm für die besondere Anwendung zu definieren. Das Steuerprogramm wird interaktiv dadurch erzeugt, daß aus Ikonen, die Teile darstellen, welche innerhalb der verschiedenartigen Steuerungsmodule 12 verfügbar sind oder zu den verschiedenartigen Steuerungsmodulen 12 herabladbar sind, ein Schemabild konstruiert wird. Nachdem das Programm vervollständigt ist, wird es in Abschnitte aufgeteilt, die den einzelnen Steuerungsmodulen 12 zugeordnet werden können, worauf hin das Programmiergerät 20 abgetrennt und das Programm ohne zentralisierte Koordination der verschiedenartigen Steuerungsmodule 12 ausgeführt werden kann.

Der erste Schritt des Programms ist der Hardwarekonfigurationsmodus, bei dem die notwendigen Teile und/oder Module festgelegt oder ausgewählt werden. Zwei prinzipielle Konfigurationsverfahren kann man in Abhängigkeit davon anwenden, ob eine on-line- oder eine off-line-Programmierung angewendet wird, was der Anwender entscheiden kann.

Bei der on-line-Programmierung, die generell durch die Prozeßblöcke 140 und 142 dargestellt ist, werden zunächst besondere Steuerungsmodule 12 ausgewählt und am Kommunikationsmittel 14 installiert, und für die Programmierung sind nur diejenigen Teile verfügbar, die durch diese Steuerungsmodule implementiert werden können. Jedes Steuerungsmodul 12 erhält eine getrennte eindeutige Adresse mittels seines DIP-Schalters 116. Die Programmierung fragt dann das Kommunikationsmittel 14 ab, wie es durch den Prozeßblock 140 angezeigt ist, um die besonderen Steuerungsmodule 12 festzustellen, die an das Kommunikationsmittel 14 angeschlossen sind, und um ihre Eigenschaften zu ermitteln.

Zum Ausführen der Anfrage kann das Programmiergerät 20 sequentielle Nachrichten an die Steuerungsmodule 12 aussenden, wobei jede Nachricht eine sequentiell unterschiedliche Adresse hat, bis alle möglichen Adressen identifiziert worden sind. Wie es weiter unten beschrieben wird, sind die Steuerungsmodule so programmiert, daß sie nur auf solche Nachrichten ansprechen, die an ihre eigene Adresse gerichtet sind. Nach Empfang einer solchen richtig adressierten Nachricht antwortet das Steuerungsmodul 12 mit seiner Modul-ID 128, seiner Ikonengrafik 130, seinen Logiküberprüfungsregeln 132, seiner Anschlußliste 134, seiner Teilelogik 104 und irgendanderen erforderlichen Informationen. Auf diese Weise werden die Teile in einem "Teilemagazin" identifiziert, wobei es sich um Teile handelt, mit denen der Programmierer arbeiten kann, um das erforderliche Steuerprogramm zu erstellen. Nach



Vervollständigung der Abfrage liefert das Programmiergerät 20, bei einem Prozeßblock 142, eine Anzeige über mögliche Netzwerkfehler, beispielsweise für den Fall, daß zwei oder mehrere Steuerungsmodule 12 dieselbe Adresse haben.

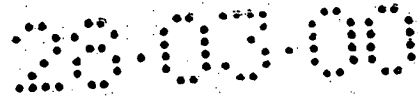
Alternativ, bei dem zweiten, dem off-line-Konfigurationsverfahren, wie es durch Prozeßblöcke 144 und 146 wiedergegeben wird, kann die Programmierung ohne die Verbindung irgendeines Steuerungsmoduls 12 mit dem Kommunikationsmittel 14 vorgenommen werden, wobei für die Programmierung alle Teile zur Verfügung stehen, die dem Programmiergerät 20 bekannt sind. In diesem Fall wird der Programmierer aufgefordert, wie es durch den Prozeßblock 144 dargestellt ist, die gewünschten Typen der Steuerungsmodule 12 in das Programmiergerät 20 einzugeben, und die erforderlichen Programmier- und Teileattribute für diese Steuerungsmodule 12 werden aus einer Moduldatenbibliothek gewonnen, die beispielsweise auf einer Magnetplatte im Laufwerk 30 vorgesehen ist. Aufgrund dieser Auswahl kompiliert das Programm zunächst, beim Prozeßblock 146, eine Teileliste der zur Programmierung verfügbaren Teile.

In beiden Fällen stellt das Programmiergerät, bei einem Prozeßblock 148, eine Anzeige über die Module 12 zur Verfügung, die im Programm benutzt werden können.

Bei einer Variante des off-line-Konfigurationsverfahrens kann der Anwender, beim Prozeßblock 144, anstelle der Module 12 Teile als solche auswählen. Verschiedenartige Module 12, die die gewünschten Teile tragen, werden dann dem Anwender präsentiert. Bei dieser Ausführungsform muß sich der Anwender unmittelbar keine Gedanken über die Implementierung der Teile innerhalb tatsächlicher Module 12 machen.

Unter zusätzlicher Bezugnahme auf Fig. 10 werden in allen Fällen die verfügbaren Teile in einem Teilemagazinbereich 150 auf dem Sichtschirm 22 des Programmiergeräts 20 dargestellt. Das Teilemagazin ist der am meisten links liegende Abschnitt des Schirms.

Die Modulikonen 152 repräsentieren im allgemeinen die äußeren Aspekte der Steuerungsmodule 12 ohne Angabe der Teile innerhalb der Steuerungsmodule 12. Eine Modulikone 152 kann vom Programmierer durch Auswahl über das Tastenfeld 26 oder die Maus 28 "geöffnet" werden, um die verschiedenartigen Teile darzulegen, die im Steuerungsmodul 12 enthalten sind, wie es durch Teileikonen 154 dargestellt ist. Jede Teileikone 154 ist durch ein



Funktionssymbol dargestellt, das eine oder mehrere Eingangs- und Ausgangsanschlüsse hat, wie es in Fig. 5 veranschaulicht ist.

Das Programm kann dann als nächstes zu einem off-line-Editiermodus voranschreiten, bei dem die dargestellten Teile innerhalb eines Programmierbereiches 168 auf dem Sichtschirm 22 zur Rechten des Teilemagazins 150 angeordnet und miteinander verbunden werden können. Jede der Stufen des Editiermodus können beliebig oft wiederholt werden, bis das Editieren vollendet ist.

Nachdem ein Steuerungstransformator 53' und zugeordnete Leistungsschalter 55 und 57 ausgewählt worden sind, wird der Programmierbereich 168 durch eine vertikale Energiezuleitungsschiene zur Linken, die eine Energiequelle darstellt, und eine vertikale Energierückleitungsschiene 80 zur Rechten, die eine Energiesenke darstellt, flankiert. Falls der Anwender die editierte Logik der Teile nicht ausführen möchte, kann das Leistungsschalterpaar 55, 57 geöffnet bleiben.

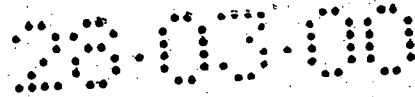
In einer Stufe des Editierprozesses kann man die Teilelogik für die Teile im Teilemagazin 150 ändern. So kann man beispielsweise in einem Schalterteil die normalerweise geschlossenen Kontakte des Schalters in normalerweise geöffnete Kontakte ändern. Eine Teilelogikänderung kann man durch die Maus 28 implementieren, wobei ein Cursor auf ein Auswahltool 160 (Abbildung eines Pfeiles) auf dem Sichtschirm 22 bewegt wird. Dieses Auswahltool 160 ersetzt den Cursor und wird zu der besonderen Teileikone 154 bewegt, deren Logik geändert werden soll, so daß die Teileikone 154 hervorgehoben wird. Zur Änderung der Teilelogik wird dann eine Änderungslogikschaltfläche 162 durch die Maus 28 aktiviert, oder es wird eine Auswahl auf einem Balkenmenu vorgenommen.

Wird eine Teilelogikänderung gewünscht, wie es durch einen Entscheidungsblock 156 festlegbar ist, kann das Programm Programmiernachrichten, die später an die Steuerungsmodule 12 auszusenden sind, in der Teilelogiktafel 122 des nichtflüchtigen RAM's 98 kompilieren, wie es durch einen Prozeßblock 158 dargestellt ist. Die im ROM 94 abgelegte Teilelogiktafel 104 umschreibt die verschiedenartigen Änderungen in der Logik, welche für ein gegebenes Teil zulässig sind. Hat sich die Logik eines Teils geändert, ändert sich die Teileikone 154, die im Teilemagazin 150 oder Schemabild 168 dargestellt ist, um die geänderte

Funktionalität wiederzuspiegeln. Das Einstellen "Anwender gewünschter Menu-Standardstellungen" wird auf diese Weise vorgenommen.

Bei einer anderen Stufe des Editiermodus, wie es beim Prozeßblock 164 wiedergegeben ist, werden die verschiedenartigen Teileikonen 154 an verschiedenen Stellen im Programmierbereich 168 positioniert. Diese Positionierung von Ikonen wird mit der Maus 28 nach Auswahl eines Lokationstools 166 vorgenommen, das den mausgesteuerten Cursor 161 auf dem Sichtschirm 22 ersetzt und verwendet wird, um die benötigte Teileikone 154 zu einem gewünschten Platz im Programmierbereich 168 des Sichtschirms 22 zu "ziehen". Es gibt keine Beschränkungen, wo die besondere Teileikone 154 innerhalb des Programmierbereichs 168 plaziert werden kann, so daß die Plazierung der Teileikonen 154 konform mit den Regeln der konventionellen Schemabildpraxis sein kann. Weiterhin kann jede Teileikone 154 innerhalb eines gegebenen Steuerungsmoduls 12 an verschiedenen Stellen innerhalb des Programmierbereichs 168 plaziert werden, um der Lesbarkeit und einem leichteren Verständnis Rechnung zu tragen. In Abhängigkeit von dem ausgewählten Teil oder aufgrund einer Definition durch den Anwender können, in einem Prozeßblock 174, Plazierungsregeln aufgerufen werden, um diesen Prozeß zu unterstützen. Die Programmierfläche 168 kann vertikal und horizontal "gerollt" werden, wie es allgemein bekannt ist, um einen im wesentlichen unbegrenzten Raum für Teileikonen 154 vorzusehen. Das Rollen der Programmierfläche 168 teilt die Programmierfläche 168 in "Blätter" auf, wobei die Bezugszahl 184 ein gegebenes Blatt bezeichnet, das als Beispiel in der oberen rechten Ecke der Programmierfläche 168 dargestellt ist.

Im Editiermodus, bei dem die Teileikonen 154 des Teilemagazins 150 in der Programmierfläche 168 plaziert werden, können die Anschlüsse der positionierten Teileikonen miteinander oder mit der Energiezuleitungs- oder Energierückleitungsschiene 74 bzw. 80 verbunden werden, und zwar wie elektrische Symbole auf einem Schemabild. Diese Verbindung kann durch Auswahl eines Verdrahtungstools 172 (Abbildung eines Verdrahtungswerkzeugs) für den Cursor 161 vorgenommen werden und durch Bewegen des Cursors über die Maus 28, und zwar der Reihe nach zu jedem Paar Anschlüsse der miteinander zu verbindenden Teileikonen. Die Verbindungen werden auf dem Sichtschirm 22 durch Verdrahtungslinien zwischen den verbundenen Anschlüssen angezeigt. Während die Verbindungen gemacht werden, wird jede Verbindung mit den vom ROM 94 hochgeladenen Logiküberprüfungsregeln 132 (Fig. 7) verglichen, um sicherzustellen, daß richtige Verbindungen gemacht werden und, in



nahezu Echtzeit, der Programmierer über irgendwelche Programmierfehler unterrichtet wird. Dieser Prozeß der Verdrahtungsüberprüfung ist durch einen Prozeßblock 174 wiedergegeben. Verbindungen können dadurch verschoben oder aufgehoben werden, daß das Verdrahtungstool 172 über die nichtgenehme Verbindung gebracht wird.

Anschlüsse 60 irgendeiner gegebenen Teileikone können mit Ikonen auf anderen Blättern dadurch verbunden werden, daß unter Verwendung des Verdrahtungstools 172 die erste Verbindung auf dem ersten Blatt gemacht wird und dann zu einem anderen Blatt geblättert wird und dort die zweite Verbindung erstellt wird. Visuell werden diese Verbindungen auf dem Sichtschirm durch eine Linie dargestellt, die zu einem Symbol führt, welches das Blatt und die Zeilennummer anzeigt, wo man das andere Ende des Drahtes finden kann.

Die Schritte des Änderns der Teilelogik gemäß dem Prozeßblock 158, des Plazierens der Teileikonen 154 auf dem Schirm gemäß Prozeßblock 164, des Verbindens der Anschlüsse gemäß dem Prozeßblock 170 und des Überprüfens der Verbindungen gemäß dem Prozeßblock 174 können wiederholt werden, bis das gewünschte Steuerungsschema erhalten wird. Dies wird generell durch eine Schleife angezeigt, die innerhalb des Entscheidungsblocks 176 ausgebildet ist, der anfragt, ob der off-line-Editierprozeß vollendet ist, und falls nicht, die Programmierung zurück zur Oberseite des Prozeßblocks 156 gebracht wird, um die Operationen der Prozeßblöcke 164, 170 oder 174 in beliebiger Folge vorzunehmen.

Das auf diese Weise erzeugte Schemabild gibt die gewünschte Logik des HDIC-Systems 10 in Form eines Satzes von Drahtverbindungen zwischen Teilen wieder, die eine besondere Teilelogik haben. Das Programm für das HDIC-System 10 ist dann in erster Linie das Schemabild, und die Programmiersprache des HDIC-Systems 10 ist in Form eines Satzes Anweisungen vorgesehen, d. h., bei dem beschriebenen Beispiel, liegen standardmäßige elektrische Teile vor, wie beispielsweise Schalter und Lampen, sowie die Verdrahtung zwischen diesen Teilen. Die Anwendung dieser Programmier-"Sprache", bei der die Logik in standardmäßigen elektrischen Teilen und deren Verbindungen untereinander enthalten ist, führt zu dem Vorteil, daß gut verständliche und hoch entwickelte elektrische Schemastandards angewendet werden können, wie sie beispielsweise verbreitet werden durch JIC, NEMA, IEC und anderen, und daß weiterhin offensichtlich eine einfache und intuitive Lösung für das Problem geschaffen wird, wie in effizienter Weise ein Programm unter vielen verteilten Prozessoren

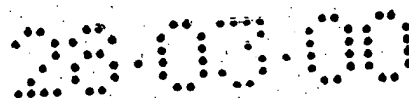


zugeordnet werden kann, wie es beispielsweise durch jeden Steuerungsmodul 12 repräsentiert ist.

Während die Anschlüsse der Teileikonen 154 verbunden werden, wie es durch den Prozeßblock 170 angezeigt ist, wird eine Drahtliste erzeugt, die letztlich in die Drahtlistentabelle 124 in den nichtflüchtigen RAM 98 herabgeladen wird. Die Drahtliste kann man unter Verwendung verschiedenartiger Zuordnungsregeln gewinnen. Dieses Beispiel unterstellt, daß die ausgewählte Regel eine sequentielle ganzzahlige Drahtnummer für jede Drahtverbindung zwischen Anschlüssen der Teileikonen 154 vorsieht, um in einfacher Weise diesen Draht zu identifizieren. Die Drahtnummern sind unabhängig von der tatsächlichen physischen Implementation sowie den verwendeten Produkten und hängen vielmehr von dem Ort der Ikonen jedes Teils auf dem Sichtschirm 22 ab, was für den Programmierer angenehm ist. Die nachstehende Tabelle I zeigt eine Drahtliste für das HDIC-System 10 von Fig. 1, 2 und 6 und assoziiert eine Drahtnummer mit Paaren von Nachrichtenidentifizierungen. Die Drahtnummer wird, abgesehen von Brücken (dies ist noch zu beschreiben), auf der Grundlage der Position auf dem Schirm zugeordnet.

Tabelle I

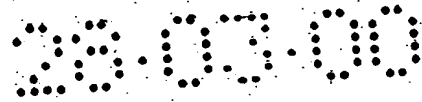
Drahtnummer	Stromaufwärtsanschluß	Stromabwärtsanschluß
100	01-3	02-0, 04-2
101	02-0, 04-2	02-1
102	02-1	03-0
103	03-0	04-0
104	04-0	04-1, 02-2
105	04-1	01-3
106	04-0	01-3



Es sei bemerkt, daß Drahtnummern nur für Verbindungen zwischen Teilen erzeugt werden, die separat miteinander verbunden werden können. Die Drahtnummer 101, die zwei Stromaufwärtsverbindungen hat, implementiert eine "verdrahtete" Oder-Funktion, die jeden ihrer beiden Stromaufwärtspfade anspricht. Sobald ein off-line-Editiervorgang beendet ist, wie es durch einen Entscheidungsblock 176 festgestellt werden kann, tritt das Programm in einen Kompiliermodus ein. In der ersten Stufe der Kompilierung, die durch einen Prozeßblock 206 dargestellt ist, überprüft das Programm jedes Teil in der Programmierfläche 168 und identifiziert die Verbindungen jedes Stromaufwärts- oder Eingangsanschlusses zu Stromabwärts- oder Ausgangsanschlüssen von Teilen von anderen stromaufwärts gelegenen Steuerungsmodulen 12. Die Anschluß- und Modulnachrichtenidentifizierungen dieser Verbindungen werden verwendet, um die Stromaufwärtsverbindungsliste 118 für das dieses Teil enthaltende Steuerungsmodul 12 zu erzeugen. Gleichmaßen werden die Stromabwärtsverbindungen jedes Stromabwärts- oder Ausgangsanschlusses eines Teils mit den Stromaufwärts- oder Eingangsanschlüssen von Teilen von anderen Steuerungsmodulen 12 als Option gesammelt, um die Stromabwärtsverbindungsliste 120 für das dieses Teil enthaltende Steuerungsmodul 12 zu erstellen.

Die Stromaufwärts- und Stromabwärtsverbindungslisten 118 und 120 jedes Steuerungsmoduls 12 werden über das Kommunikationsmittel 14 zu den jeweiligen Steuerungsmodulen 12 herabgeladen, wie es durch einen Prozeßblock 208 angezeigt ist. Gleichmaßen werden Programmiernachrichten, die in der oben beschriebenen Weise im Prozeßblock 158 erzeugt werden und Änderungen in der Logik der besonderen Teile angeben, unter Ausbildung der in Fig. 7 dargestellten Teilelogiktable 122 herabgeladen. Die Drahtlisten 124 und Kennungen 126, die während der Programmierung des HDIC-Systems 10 erzeugt wurden, werden ebenfalls herabgeladen. Dieses Herabladen kann nach Vollendung der Editierung vorgenommen werden, wie es dargestellt ist, oder während des Editierens, falls dies erwünscht ist.

Mit dem Abschluß des Editierens kann man das HDIC-Programm durch Schließen des Leistungsschalterpaares 55, 57 laufen lassen. Dadurch wird die Verbreitung eines Laufbefehls längs des Kommunikationsmittels 14 ausgelöst, wie es durch einen Prozeßblock 210 angezeigt ist. Das System kann auch im Betrieb oder während irgendeiner Kombination aus in



Betrieb befindlichen Modulen oder Ikonen oder vollständig off-line (mit offenen Leistungsschaltern) editiert werden, wie es oben beschrieben worden ist.

### Eine Editiersitzung

Es wird auch auf Fig. 10 und 11 bezug genommen. Für das in Fig. 1 dargestellte HDIC-System 10 kann das Schemabild nach Fig. 11 auf dem Sichtschirm 22 aus Steuerungsmodulen 46, 40, 42 und 52 erzeugt werden, die im Teilemagazin 150 durch Modulikonen dargestellt sind. Anfangs kann man die Druckastenmodulikone öffnen, um die Druckastenteile einschließlich der normalerweise geöffneten Druckastetikone 180, die der Druckaste 38 entspricht, freizulegen. Die Maus 28 und das Lokationstool 166 (Abbildung einer Hand) werden dann verwendet, um die Druckastetikone 180 aus dem Teilemagazin 150 in den Programmierbereich 168 zu verschieben. Die Druckaste 180 kann irgendwo im Programmierbereich 168 platziert werden, jedoch wird wegen ihrer logischen Beziehung zu anderen Teilen von anderen Steuerungsmodulen 12 und der Konvention des Stromflusses von der Energiezuleitungsschiene 74 zur Energierückleitungsschiene 80 die Druckastetikone 180 auf der linken Seite des Programmierbereiches 168 nahe bei der Energiezuleitungsschiene 74 positioniert.

Sobald die Ikone 180 positioniert ist, kann ihr am weitesten links liegender Anschluß unter Verwendung des Verdrahtungstools 172 mit der Energiezuleitungsschiene 74 verbunden werden, wie es oben beschrieben worden ist. Sobald, unter Bezugnahme auf Fig. 11, die Verbindung zwischen dem am weitesten links befindlichen Anschluß der Druckastetikone 154 gemacht ist, erscheint eine diese Verbindung anzeigende Linie mit der Drahtnummer 100 unmittelbar zur Rechten der Energiezuleitungsschiene 74. Die Verbindung des linken Anschlusses der Druckastetikone 180 mit der Energiezuleitungsschiene 74 verursacht auch, daß die Druckaste mit einer Kennung versehen wird, in diesem Fall mit 1LPB (erste beleuchtete Druckaste). Die Numerierung der Drahtverbindungen und Teile folgt irgendeiner verschiedenartiger genormter Industriepraktiken für Schemabilder und kann gemäß der Position der Verbindung und des Teils auf der schematischen Bilddarstellung (Fig. 11), die mit der Bezugszahl 182 gekennzeichnet ist, erfolgen.

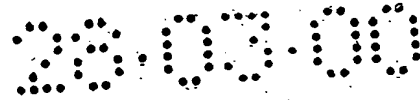


Bei der weiteren Programmierung des HDIC-Systems 10 für die Anwendung nach Fig. 1 und 2 kann man die Teileikone 186 (Fig. 11), die den normalerweise geöffneten Kontakt 76 im Motorstartmodul 52 darstellt, so verbinden, daß sie parallel zu der Druckastenikone 180 geschaltet ist. Damit kann man, immer noch unter Bezugnahme auf Fig. 11, die Ikone 186 für die normalerweise geöffneten Hilfskontakte des Motorstartmoduls 52 zu einer Position unterhalb der Ikone 180 der beleuchteten Druckaste ziehen, und das Verdrahtungstool 172 wird ausgewählt, um den linkesten Anschluß der Ikone 186 mit der Energiezuleitungsschiene 74 zu verbinden.

Als nächstes wird die Teileikone 188 der normalerweise geschlossenen Druckaste 44 des Druckastenmoduls 40 auf die rechte Seite der Teileikone 180 geschoben, und der Stromaufwärtsanschluß dieser Ikone 188 wird mit dem rechtesten Anschluß der Ikone 180 mittels des Verdrahtungstools 172 verbunden. Gleichzeitig kann man das Verdrahtungstool 172 dazu verwenden, um den rechtesten Anschluß der Ikone 186 mit dem linkesten Anschluß der Ikone 188 zu verbinden.

Das Programmierschema von Fig. 11 kann man zur Konstruktion weiterführen, und zwar dadurch, daß als nächstes die Teileikone 192 ausgewählt wird, die dem Grenzschaaltermodul 42 entspricht, und diese stromabwärts der Ikone 188 verbindet. Wenn diese Ikone verbunden wird, erhält sie die Kennung "ILS", was erster Grenzschaalter bedeutet. Die Teileikone 194 wird durch iterative Auswahl von drei normalerweise geschlossenen Kontakten 78 konstruiert, und dann wird deren linkester Anschluß mit dem rechtesten, normalerweise geschlossenen Anschluß der Ikone 192 verbunden. Eine Teileikone 196, die die Spule 72 des Motorstarters darstellt, wird zwischen den rechtesten Anschluß der drei normalerweise geschlossenen Kontakte 78 und die Energierückleitungsschiene 80 geschaltet, wobei eine Sprosse gebildet wird. Letztlich wird noch eine Teileikone 200, die die Lampe 39 der beleuchteten Druckaste darstellt, so verbunden, daß sie die Aktorikone 196 kurzschließt, was bedeutet, daß ihr linkester Anschluß mit dem rechtesten Anschluß der drei normalerweise geschlossenen Kontakte der Ikone 194 verbunden ist. Der rechteste Anschluß der Lampe 39 ist mit der Energierückleitungsschiene 80 verbunden. Jetzt ist die Programmierung abgeschlossen.

In den Editiermodus kann man zu jeder Zeit wieder eintreten, und man kann ihn zur Fehlersuche benutzen. So kann beispielsweise der Programmierer ein Kraft-aus-Tool (Abbild eines Drahtschneiders) oder ein Kraft-ein-Tool 204 (Abbild einer Brücke) verwenden, um



zeitweilige Verbindungen zu erstellen und zu markieren. Zur Untersuchung des Programms auf Fehler kann es beispielsweise erwünscht sein, die Lampe 39 der beleuchteten Drucktaste zeitweilige direkt mit der Energiezuleitungsschiene 74 und der Energierückleitungsschiene 80 zu verbinden. Diese Verbindung würde erhellen, daß die Lampe in Wirklichkeit arbeitet. Unter Beachtung der oben beschriebenen Konvention kann man die Verbindung durch Verwendung des Kraft-ein-Tools 204 errichten, wobei die Energiezuleitungsschiene 74 mit dem linken Anschluß der Lampe 39 verbunden wird. Diese Verbindung würde durch eine bogenförmige gelbe Linie anstelle einer geradlinigen roten Linie auf dem Sichtschirm 22 angezeigt werden. Im Anschluß an diese Verbindung kann man, anstatt die einen Draht darstellende Linie, durch Verwendung des Verdrahtungstools 172 zu entfernen, irgendeinen Draht unter Verwendung des Trenn-Tools 202 auftrennen. Dies hinterläßt zwei lose Drahtsymbole 205, die durch eine gepunktete Linie miteinander verbunden sind, um für eine mögliche zukünftige Verwendung den Ort der Brücke zu markieren, die benutzt wurde, um die beleuchtete Drucktaste zwangsmäßig einzuschalten.

Wiederum unter Bezugnahme auf Fig. 10 sei erläutert, daß irgendeiner der Moden der Konfiguration, des Editierens, des Kompilierens und des Laufens durch Aktivieren einer visuellen "Taste" aufgerufen werden kann, die oberhalb der Programmierfläche 168 dargestellt ist. Die Aktivierung einer solchen Taste veranlaßt, daß das Programm bezüglich dieses Modus zu den vorstehend beschriebenen Prozeßblöcken voranschreitet.

### Betrieb der Steuerungsmodule

Unter Bezugnahme auf Fig. 4, 7 und 9 läuft in jedem Steuerungsmodul 12 nach Empfang von Energie vom Kommunikationsmittel 14 ein Programm ab, wie es in seinem ROM 94 gespeichert ist. Aus Gründen der Klarheit ist das Programm in Form einer kontinuierlichen Schleife dargestellt, jedoch ist es einem Fachmann klar, daß andere Programmarchitekturen auch benutzt werden können, um die hier beschriebenen Funktionen zu bewerkstelligen.

Falls die Verbindungs- oder Bindesteuerung 86 in Form einer separaten Schaltung vom Mikroprozessor 100 realisiert wird, dann überwacht die Bindesteuerung 86 gleichzeitig mit diesem Programm Nachrichten auf dem Netzwerk und speichert, auf Unterbrechungs- oder Interruptbasis, relevante Nachrichten in einem Pufferbereich des flüchtigen RAM 96.

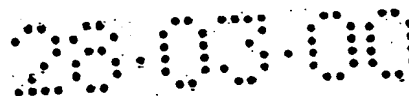


Das Programm überwacht periodisch den volatilen RAM 96 auf gespeicherte Nachrichten, die vom Kommunikationsmittel 14 empfangen worden sind, wie es durch einen Prozeßblock 220 angezeigt ist. Zu diesem Zeitpunkt wird auch der Status irgendeines Operators 108, beispielsweise einer Drucktaste, festgestellt und als ein Merker im volatilen RAM 96 gespeichert.

Bei einem Entscheidungsblock 244 wird irgendeine gespeicherte Nachricht überprüft, um festzustellen, ob es sich um eine Systemanforderungsnachricht mit hoher Priorität (die beispielsweise einen "Komplettstop" vom Modul 12 fordert, der dann seine Steuerwirkung stoppt) handelt, wie es durch einen eindeutigen Nachrichtenkopf (ein Arbitrationsfeld) angezeigt wird, und ob es sich um eine eingebettete Nachrichtenidentifizierung handelt, die mit der Moduladresse dieses Steuerungsmoduls 12 übereinstimmt. Die Anforderung bezeichnet auch die spezifischen Aktionen, die vom Steuerungsmodul gefordert werden, beispielsweise seine Rücksetzung.

In jedem Falle schreitet dann das Programm zu einem Entscheidungsblock 226 voran, bei dem die gespeicherte bzw. zwischengespeicherte Nachricht daraufhin überprüft wird, ob es sich um eine Produktionsnachricht mit einer Nachrichtenidentifizierung auf der Stromaufwärtsverbindungsliste 118 handelt. Produktionsnachrichten sind Ausgangsnachrichten von anderen Steuerungsmodulen 12, mit denen ein Teil des empfangenden Steuerungsmoduls 12 virtuell verbunden ist. Ist das Steuerungsmodul 12 beispielsweise das Drucktastenmodul 40, würde das Drucktastenmodul 40 auf eine Nachricht von dem Teil, das den Kontakt 76 im Motorstartmodul 52 darstellt, reagieren. In diesem Fall würde der Operatorstatus, d. h. der Status der Drucktaste, mit Hilfe des Merkers im volatilen RAM 96 überprüft werden, wie es durch den Prozeßblock 228 gezeigt ist.

Als nächstes aktualisiert das Programm bei einem Prozeßblock 230 auf der Grundlage der besonderen Produktionsnachricht und dem Zustand des Operators und unter Bezugnahme auf die Teilelogik im RAM 94 einen Ausgangszustand für dieses Teil, der den Wert eines Stromabwärtsanschlusses dieses Teils darstellt, und speichert diesen Zustand in einer Tabelle im volatilen RAM 96. Während der Aktualisierung der Tabelle wird bei einem Entscheidungsblock 232 ihr vorangegangener Inhalt überprüft, um festzustellen, ob sich bei der Aktualisierung der gegebene Ausgang geändert hat, und falls dies zutrifft, sendet das Steue-

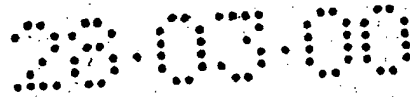


rungsmodul 12 bei einem Entscheidungsblock 234 seine eigene Produktionsnachricht oder seine eigenen Produktionsnachrichten mit seiner eigenen Nachrichtenidentifizierung aus, und zwar zugunsten von anderen stromabwärtsgelegenen Steuerungsmodulen 12. Eine Produktionsnachricht wird nur dann ausgesendet, wenn sich der Ausgang geändert hat, und nicht, wenn sich lediglich der Operatorzustand geändert hat, um auf diese Weise die Kapazität des Netzwerks aufrecht zu erhalten. Unter Bezugnahme auf einen Prozeßblock 246 muß diese Aussendung nicht zu diesem Zeitpunkt erfolgen, sondern braucht lediglich eingeplant zu werden, und zwar zusammen mit der Handhabung von anderen Kommunikationstasks zu einer späteren Zeit.

Dennoch ist es für einen Fachmann selbstverständlich, daß bei ausreichender Netzwerkkapazität eine Ausgangsnachricht immer ausgesendet werden kann. Weiterhin wird, wenn sich der Ausgang nicht geändert hat, sondern das Steuerungsmodul 12 in einem Diagnostikmodus ist, was durch einen Merker festgelegt ist, der beim Prozeßblock 224 gesetzt wird, eine Produktionsnachricht ausgesendet. Im Diagnostikmodus werden die Ausgangszustände fortlaufend auf das Kommunikationsmittel 14 ausgesendet, so daß die Anzeige auf dem Bildschirm 22 fortlaufend aktualisiert und eine Echtanzeige über den Betrieb des gesamten HDIC-Systems 10 geliefert wird. Diese Lösung hilft auch bei der Emulation einer den schlimmsten Fall betreffenden Netzwerkbelastung im tatsächlichen System.

Diese Schritte 226 bis 234 repräsentieren den Kern des Betriebs jedes Steuerungsmoduls 12 und verlangen von dem Steuerungsmodul lediglich, daß das Modul die Nachrichtenidentifizierungen von irgendwelchen stromaufwärtsgelegenen Teilen identifiziert, die mit dem Steuerungsmodul 12 verbunden sind. Das Steuerungsmodul 12 ist bezüglich seiner Primärfunktion hinsichtlich der stromabwärtsgelegenen virtuellen Verbindungen indifferent.

Der Abschnitt des industriellen Steuerprogramms, der jedem einzelnen Steuerungsmodul 12 zugeordnet werden muß, ist deshalb hauptsächlich ein Satz aus stromaufwärts Nachrichtenidentifizierungen, die die virtuellen Verbindungen vom Steuerungsmodul 12 zu den stromaufwärts Steuerungsmodulen 12 erstellen. Diese virtuellen Verbindungen und die Teilelogik, die in der ROM-Tabelle 104 enthalten ist, bilden das Programm, das von jedem Steuerungsmodul 12 ausgeführt wird.



Beim Entscheidungsblock 222 wird der Pufferbereich des RAM 96 überprüft, um festzustellen, ob eine "Kunden"-Nachricht empfangen worden ist. Trifft dies zu und stimmt die Adresse der Kundennachricht mit der Knotenadresse des empfangenden Steuerungsmoduls 12 überein, bedient das Steuerungsmodul 12 die Kundenanforderung.

Eine Kundennachrichtanforderung bedeutet das Heraufladen von persönlichen Daten aus dem nichtvolatilen RAM 98 oder dem ROM 94 des Moduls. Bei einer Ausführungsform können Ikonengrafiken 130 und die Logiküberprüfungsregeln 132 als auch die Anschlußliste 134 und die Modul-ID 128 heraufgeladen werden. Der Inhalt des nichtvolatilen RAM's 98 einschließlich der Drahtliste 124 und der Kennung 126 kann für die Zwecke der Wiederherstellung des Programms, das am hochverteilten Netzwerk arbeitet, ebenfalls heraufgeladen werden, falls der ursprüngliche Source- oder Quellcode nicht verfügbar ist und rekonstruiert werden muß. Als Antwort auf eine gültige und richtig adressierte Heraufladeanforderungsnachricht, was bei der Entscheidung 244 überprüft wird, lädt das Steuerungsmodul 12 die angeforderten Daten, bei einem Prozeßblock 246, herauf auf das Kommunikationsmittel 14. Dieser Heraufladevorgang muß zu diesem Zeitpunkt nicht auftreten, sondern kann lediglich in einfachster Weise in Verbindung mit der Handhabung von anderen Kommunikationstasks zu einem späteren Zeitpunkt geplant werden.

Ein Beispiel einer anderen Kundennachricht ist eine Programmieranforderung, die den Inhalt der Stromaufwärtsverbindungsliste 118, der Stromabwärtsverbindungsliste 120, die Teilelogiktafel 122 oder Drahtliste 124 oder die Kennungen 126 zu diesem Modul 12 herablädt, wie zuvor beschrieben. Dieses Herabladen zu dem Steuerungsmodul 12 wird durch einen Prozeßblock 224 angezeigt. Beim Prozeßblock 224 kann das Steuerungsmodul 12 auch eine Fehlersuchstatusnachricht empfangen, die im volatilen RAM 96 ein Diagnostikmodusmarkierbit setzt, das in einer weiter unten zu beschreibenden Weise benutzt wird.

Wie erwähnt, werden bei der bevorzugten Ausführungsform Produktionsnachrichten nur dann übertragen, wenn sich der Ausgang eines Teils in dem Steuerungsmodul 12 ändert. Wegen Steuerungszwecken wird eine Nachricht nur zu einer Zeit erzeugt, wenn derjenige Ausgang des Moduls beeinträchtigt wird, der Stromabwärtsmodule speist. Diese Prozedur konserviert Netzwerkkapazität, birgt jedoch die Gefahr in sich, daß ein fehlerhaftes Steuerungsmodul 12 nicht erfaßt wird. Verschiedenartige Methoden, wie beispielsweise Nachrichtenüberwachung, kann man einsetzen, um gestörte Module und/oder Programmlogik zu erfassen.

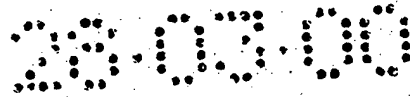


sen. Um aber, die Modulkomplexität weiter zu verringern, wird die Fehlererfassung und Fehleraktion bei der bevorzugten Ausführungsform ausschließlich von Leistungsschaltern, Aktoren oder Ausgangsmodulen vorgenommen, wie es für das Trennmodul 46 zutrifft. Die Fähigkeit eine zwingende Berichtsnachricht pro Prozeßblock 241 zu generieren, wird daher lediglich durch einige Steuerungsmodule 12 implementiert.

Von diesen Modulen 12, denen ein Zwangsbericht implementiert ist, wird eine Zwangsberichtsauflösenachricht an alle Module 12 in einem vorbestimmten Intervall ausgesandt, wie es durch den Prozeßblock 241 dargestellt ist. Das Intervall zwischen solchen Auflösenachrichten ist hinreichend lang gewählt, so daß durch diese zusätzlichen periodisch wiederholten Aussendungen die Netzwerkskapazität nicht unnötig belastet wird.

Als Antwort auf die Zwangsberichtsauflösenachricht senden Teile in Modulen 12, die mit der Energiezuleitungsschiene 74 verbunden sind, eine Zwangsberichtsnachricht an ihre Stromabwärtsnachbarn aus, welche in ihren Stromabwärtsverbindungslisten 120 aufgeführt sind. Eine solche Nachricht umfaßt den derzeitigen Ausgangszustand der Teile, veranlaßt aber nicht irgendeine Änderung des Zustands der Stromabwärtssteile, wie es bei einer herkömmlichen Ausgangsnachricht der Fall wäre. Jeder Stromabwärtsnachbar, der eine richtig adressierte Zwangsberichtsnachricht empfängt, sendet auch eine Zwangsberichtsnachricht aus, so daß die Berichtsnachricht in Stromabwärtsrichtung weiterläuft.

Das Teil, von dem die Zwangsberichtsauflösenachricht ausgeht, startet auch einen Auslösezeitgeber und "lauscht" dann auf die Vollendung des Durchlaufens der Zwangsberichtsnachrichten, wie es sich im ursprünglichen Teil durch den Empfang einer Zwangsberichtsauflösung durch seine Stromaufwärtsnachbarn widerspiegelt. Bei dem Zwangsberichtsprozeß muß jeder unmittelbare Stromaufwärtsnachbar eines Teils eine Zwangsberichtsnachricht zu diesem Teil für dieses Teil aussenden, um seine eigene Zwangsberichtsnachricht zu seinen unmittelbaren Stromabwärtsnachbarn zu übermitteln. Stromaufwärtsanschlüsse eines Teils mit mehr als einer Verbindung zu anderen Teilen, deren Anschlüsse normalerweise wie ein logisches Oder arbeiten (also auf irgendeines der Stromaufwärtssteile ansprechen), arbeiten für den Zweck der Zwangsberichtsnachrichten wie ein logisches Und (sprechen nur auf alle Stromaufwärtssteile zusammen an). Wenn somit das auslösende Teil der Leistungsschalter 57 ist, der mit der Energierückleitung 80 verbunden ist, wird der Zwangs-



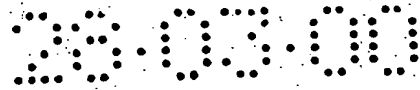
berichtet nur dann als vollständig erachtet, wenn jede Sprosse, die mit der Energierückkehrleitung 80 verbunden ist, eine Zwangsberichtsnachricht an den Leistungsschalter 57 liefert.

Wenn der Zwangsberichtsdruck vor Ablauf des Auslösezeitgebers nicht vollendet ist, wird ein Fehlerzustand angezeigt. Das ursprüngliche Teil kann dann beispielsweise dadurch aktiv werden, daß der Leistungsschalter geöffnet wird, um die Sprossen abzutrennen.

Die Zwangsberichtsnachricht gestattet es, daß der Status jedes Teils jedes Steuermoduls 12 auf einer gegebenen Bahn von der Energierückleitungsschiene 80 zu der Energiezuleitungsschiene 74 erstellt wird, so daß der Zustand des Systems aktualisiert und bestätigt wird, daß die Teile von allen Steuermodulen 12 angeschlossen sind. Dieser Status kann durch das angeschlossene Programmiergerät 20 bestätigt und dargestellt werden. Diese Technik validiert, daß alle seriellen, parallelen und rückgeführten Pfade und Bahnen intakt und ausführbar sind.

Die Entscheidungs- und Prozeßblöcke 236 bis 241 sehen somit ein Verfahren zum Schützen eines gegebenen Steuermoduls 12 gegen Fehler oder einen Bruch in seinen virtuellen Verbindungen vor. Hierbei ist es allerdings wichtig, daß ein Fehler eines gegebenen Steuermoduls 12 oder eines Teils innerhalb dieses Moduls durch dieses Verfahren leicht isoliert werden kann. Im Gegensatz dazu verhindert im allgemeinen ein Fehler einer Komponente einer zentralisierten Steuerung eine derart leichte Fehlersuche. Die einzigen Module, die eine "Überwachungsfunktion" und Fehleraktionseigenschaften benötigen, sind, wie bereits erwähnt, Ausgänge und Aktoren, aber nicht alle Netzwerkgerätschaften.

Für den Fall, daß ein gegebenes Steuermodul 12 beschädigt oder fehlerhaft ist, wie es durch das oben beschriebene Zwangsberichtsnachrichtenverfahren oder mittels seiner Statuslampe 106 oder aufgrund offensichtlicher äußerer Schadensanzeichen festgestellt worden ist, kann das beschädigte Steuermodul 12 ersetzt und das HDIC-System 10 umprogrammiert werden, um den Anteil des Programms zu ersetzen, der sich in dem beschädigten Steuermodul 12 befindet. Das Steuermodul 12, das zum Ersetzen des beschädigten Steuermoduls 12 ausgewählt worden ist, muß die geeigneten Teile tragen und kann durch ein externes Gerät, wie beispielsweise das Programmiergerät 20, hinsichtlich gewisser persönlicher Züge programmiert werden. Nachdem die Netzwerkadresse auf die Adresse des beschädigten Steuermoduls 12 eingestellt worden ist, kann die Wiederprogrammierung

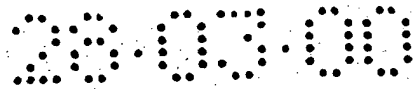


über das Programmiergerät 20 erfolgen, das das Programm (wie es auf einer Diskette gespeichert ist) einfach erneut kompiliert und dann die erforderliche Programmierinformation auf die verschiedenartigen Steuerungsmodule 12 herablädt, und zwar wie bei der ursprünglichen Programmierprozedur.

Alternativ kann die Wiederprogrammierung ohne das Programmiergerät 20 mit Hilfe der anderen Steuerungsmodule 12 vorgenommen werden, die die erforderliche Information enthalten, um das fehlende Programm in dem wieder hergestellten Steuerungsmodul 12 zu rekonstruieren. Speziell ist jedes Steuerungsmodul 12, wie es durch einen Entscheidungsblock 237 dargestellt ist, so programmiert, daß es, wenn es Produktionsnachrichten am Kommunikationsmittel 14 erfaßt und noch keine Adressen in seiner Stromaufwärtsverbindungsliste 118 oder seiner Stromabwärtsverbindungsliste 120 hat und deshalb vermutlich ein Ersatzmodul ist, eine Reihe von Reparaturanforderungsnachrichten aussendet, wie es durch einen Prozeßblock 239 angezeigt ist, die die Nachrichtenidentifizierungen der Anschlüsse seiner Teile angeben.

Wenn, wie es durch einen Entscheidungsblock 240 dargestellt ist, ein Steuerungsmodul 12 eine Reparaturanforderung mit einer Nachrichtenidentifizierung in seiner Stromaufwärtsverbindungsliste 118 oder seiner Stromabwärtsverbindungsliste 120 empfängt, dann sendet das Steuerungsmodul 12 die Nachrichtenidentifizierung seines zugeordneten Anschlusses aus, mit dem die Übereinstimmung gefunden wurde, und zwar zusammen mit der Information, ob die Übereinstimmung in der Stromaufwärtsverbindungsliste 118 oder in der Stromabwärtsverbindungsliste 120 gefunden wurde. Aus diesen Daten kann das reparierte oder wieder hergestellte Steuerungsmodul 12 seinen eigenen Stromaufwärtsverbindungsspeicher und Stromabwärtsverbindungsspeicher rekonstruieren und sich auf diese Weise selbst am Netzwerk wieder anschließen.

Das reparierte Steuerungsmodul 12 mag die verschiedenartigen Anwenderkomfortinformationen wie die Drahtliste oder die Kennung für das Steuerungsmodul 12 nicht enthalten. Diese Informationsteile werden jedoch nur dafür benötigt, um das Programm auf dem Bildschirm 22 darzustellen, und sind deshalb nicht erforderlich, um das System zurück in einen Betriebszustand zu bringen. Alternativ kann die erforderliche zusätzliche Programmierinformation, wie beispielsweise die Teilelogiktable 122, in Nachbarmodulen 12 gespeichert sein, um eine vollständige Wiederprogrammierung zu gestatten.



### Betrieb des HDIC-Systems

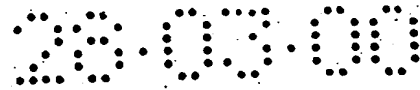
Es wird jetzt auf Fig. 11 bezug genommen. Vor dem Start des Förderbandes 32 befindet sich das Bedienelement oder der Operator der Drucktaste 38 in angehobener Stellung, so daß die Anschlüsse dieses Schalters getrennt und der Kontakt 76 geöffnet ist. Wenn die Drucktaste 38 geschlossen wird, aktualisiert das Steuerungsmodul 12, welches das Programm von Fig. 9 ausführt, direkt den Ausgang der Drucktaste, wie es gemäß der Teilelogik 104 erforderlich ist, weil der Stromaufwärtsanschluß der Drucktaste mit der Energieschiene verbunden ist. In dieser Situation heißt dies, daß bei betätigtem Drucktastenoperator sich der Ausgang notwendigerweise ändert, weil er direkt mit der Energiezuleitungsschiene 74 verbunden ist, deren Leistungsschalterkontakte geschlossen sind.

Eine Produktionsnachricht kann auf das Kommunikationsmittel 14 übertragen werden, obgleich die einzige Stromabwärtsverbindung zur Drucktaste 38 innerhalb des Moduls 40 enthalten ist. Dies reflektiert eine Philosophie, daß Steuerungsmodule 12 zu ihren Stromabwärtsnachbarn indifferent sein können, mit Ausnahme gewisser spezieller Nachrichten, wie die Zwangsberichtsnachricht oder die Reparaturanforderungsnachricht. Ferner gestattet diese Übertragung der internen virtuellen Verbindungen die Überwachung des Netzwerks durch das Programmiergerät 20.

Alternativ kann bei einer zweiten Ausführungsform vor der Übertragung irgendeiner Produktionsnachricht auf dem Kommunikationsmittel 14 das Programm eine Überprüfung vornehmen, um nachzusehen, ob sein Ausgang mit irgendeinem extern erzeugten Anschluß 60 verbunden ist. Trifft dies nicht zu, wird es die Netzwerkübertragung unterdrücken und damit beginnen, die Logik irgendwelcher verbundener Teile in diesem selben Steuerungsmodul 12 zu bewerten.

Beim Start der Fördereinrichtung befindet sich die Drucktaste 44 in einem geschlossenen Zustand, und ihr Ausgang wird daher ansteigen, und zwar mit dem Ergebnis, daß eine Produktionsnachricht auf dem Netzwerk mit der vom Grenzscharter 42 konsumierten Nachrichtenidentifizierung übertragen wird.

Die vom Modul 40 übertragene Nachricht wird vom Grenzschartermodul 42 empfangen, das, weil sich dieser Schalter in einem geschlossenen Zustand befindet, eine Produk-



tionsnachricht aussendet, die eine Änderung im Zustand an seinem erzeugten oder produzierten Anschluß anzeigt. Diese Produktionsnachricht wird von den normalerweise geschlossenen Kontakten 78 innerhalb des Motorstartmoduls 52 empfangen, das die Spule 72 erregt. Weil die normalerweise geschlossenen Kontakte 78 anfangs geschlossen sind, ändert sich ihr Ausgang gleichermaßen wie der "Zustand" vom Draht 104, wodurch eine noch andere Produktionsnachricht am Kommunikationsmittel 14 ausgelöst wird.

Diese Produktionsnachricht wird von der Lampe 39 des Drucktastensteuermoduls 40 empfangen. Die Lampe 39 ist mit ihrem Stromabwärtsanschluß mit der Energierückleitungsschiene 80 verbunden, und es wird eine Produktionsnachricht erzeugt. Durch die Aktivierung der Spule 72 wird der Kontakt 76 geschlossen, und es wird durch den Ausgang des Kontaktes 76 eine letzte Nachricht ausgelöst, die anzeigt, daß der Kontakt geschlossen ist. Da die Ankunft dieser Nachricht über den Draht 101 den Zustand des Ausgangs der Drucktaste 44 am Draht 102 nicht ändert, erzeugt das Drucktastenmodul 40 keine weitere Nachricht.

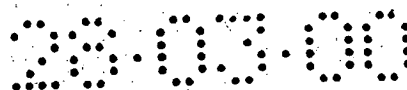
Der Betrieb des HDIC-System 10 ahmt somit die tatsächlichen Stromflüsse durch die elektrischen Standardteile nach, wie sie durch die Teile der Steuerungsmodule 12 dargestellt sind, und zwar mittels Produktionsnachrichten, die auf dem Kommunikationsmittel 14 übertragen werden.

### HDIC-Anwendung auf ein Kugelmeßsystem

#### Hardware

Unter Bezugnahme auf Fig. 12 kann eine Industriesteuerung 300 derart ausgelegt sein, daß sie die Freigabe von Kugeln 302 durch ein Rohr 304 steuert. Ein großer Vorrat von Kugeln 302 befindet sich in einem Trichter 306, der sich nach unten verengt und an seinem unteren Ende in die obere Öffnung des Rohres 304 übergeht, das den Eintritt einer einzigen Kugel 302 aus der Position (a) gestattet. Die Kugel 302 wird dann im Rohr 304 bei einer Position (b) von einer Klinke 308 festgehalten, die an einem Solenoid 310 befestigt ist.

Durch Drücken einer LOAD-Drucktaste 312 für eine vorbestimmte minimale Zeitperiode leuchtet eine Lampe 313 auf, und das Solenoid 310 wird betätigt, um die Klinke 308 aus dem Inneren des Rohres 304 zurückzuziehen, so daß die Kugel 302 im Rohr 304 von der Po-



sition (b) zu einer Position (c) herunterfallen kann, wobei die zuletzt genannte Position tiefer im Rohr 304 vorgesehen ist. Die Kugel 302 wird in der Position (c) durch eine zweite Klinke 314 festgehalten, die an einem Solenoid 316 befestigt ist.

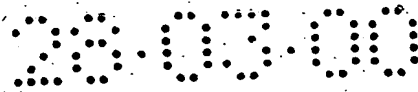
In der Position (c) oberhalb der Klinke 314 kann die Kugel 302 durch einen Näherungsschalter 318 erfaßt werden.

Wird eine REL-Drucktaste 320 (REL = Freigabe) gedrückt, leuchtet eine Lampe 322 auf und die Klinke 314 wird vom Solenoid 316 zurückgezogen, wobei ein einziges Produkt 302 freigegeben wird, sofern der Näherungsschalter 318 ein Signal abgibt, das das Vorhandensein eines Produkts 302 bei der Klinke 314 anzeigt.

Das oben beschriebene System 300 erfordert eine Anzahl elementarer Funktionen: Drucktastenschalter, Lampen, Solenoide, einen Näherungsschalter und eine Filterfunktion, die sicherstellt, daß die LOAD-Drucktaste für eine gegebene Zeitspanne gedrückt worden ist. Diesen elementaren Funktionen wird notwendigerweise nicht jeweils ein separates Modul zur Implementierung zugeordnet. Effizienz- und Kostenüberlegungen sowie räumliche Beschränkungen diktieren vielmehr, daß einige elementare Funktionen zur Implementierung durch ein einziges gegebenes Modul zu Gruppen zusammengefaßt werden.

Eingänge von menschlichen Operatoren in der Form von Drucktasten 312, 320 und Ausgänge an den Operator in Form von Lampen 313 und 322 werden aus Gründen der Bequemlichkeit für den Operator oder Bediener in logischer Weise miteinander gruppiert. Diese Funktionen werden somit durch ein Modul 350 wahrgenommen, das auf seiner Frontseite zwei Lampen 313 und 322 sowie zwei Drucktastenoperatoren für die Drucktasten 312 und 320 hat, welche Teile generell mit Schaltungsanordnungen in Verbindung stehen, wie es zuvor unter Bezugnahme auf Fig. 4 beschrieben worden ist, wodurch das Modul 350 in der Lage ist, mit dem Kommunikationsmittel 14 zu kommunizieren.

Gleichermaßen können die Solenoide 316 und 310 durch ein Modul 352, das ebenfalls mit dem Kommunikationsmittel 14 in Verbindung steht, implementiert werden, wobei die tatsächlichen Solenoide 310 und 316 über Schraubklemmen 353 angeschlossen sind, um die Solenoide 310 und 316 mit Energie zu versorgen. Die räumliche Nähe der Solenoide 310 und



316 legt es wiederum nahe, daß sie durch ein einziges Anschlußblockmodul 352 implementiert werden.

Der Näherungsschalter 318, obgleich er in räumlicher Nähe zu den Solenoiden 310 und 316 angeordnet ist, erfordert ein Modul, das Eingänge akzeptieren kann, und zwar anstelle eines Moduls, das Ausgänge erzeugen kann. Der Näherungsschalter 318 ist deshalb in einem separaten Eingangsmodul 354 implementiert, das einen Anschluß hat, der mit einer Leitung des Näherungsschalters 318 verbunden ist.

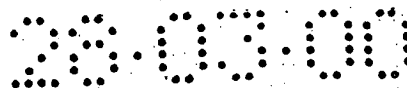
Jede der oben beschriebenen elementaren Funktionen hat eine räumliche Lage, die durch das physische Layout der Steuerungsanordnung oder durch die Notwendigkeit diktiert ist, mit einem Bediener zusammen zu arbeiten. Eine Ausnahme bildet das FLT 102, dessen sämtliche Eingänge von anderen elementaren Funktionen innerhalb des hochverteilten Steuerungssystems stammen und das deshalb willkürlich innerhalb der Module 354, 350 und 352 plaziert werden kann. In diesem Falle ist es innerhalb des Moduls 354 positioniert, und zwar gemäß dem physischen Diagramm 252, wie es weiter unten noch erläutert wird.

## Programmieren

### A. Das logische Diagramm

Unter Bezugnahme auf Fig. 13 und 14 ist der Unterschied zwischen der logischen Struktur der industriellen Steuerung 300 und ihrer physischen Struktur durch die Verwendung von zwei separaten Diagrammen eingefangen, nämlich durch ein logisches Diagramm 250 und ein physisches Diagramm 252. Im allgemeinen beschreibt das zuerst genannte die logischen Verbindungen zwischen Elementen der Steuerung, und das zuletzt genannte beschreibt die physischen oder räumlichen Verbindungen zwischen den Komponenten, die hochverteilt sind. Diese beiden Diagramme dienen als "Programmiersprache" für das hochverteilte Steuerungssystem 300.

Das logische Diagramm 250 folgt im allgemeinen der Form des Diagramms der Fig. 3, enthält aber nicht die punktierten Umrisse, die die Ansammlung der elementaren Funktionen angeben, die von den einzelnen logischen Ikonen 254 ausgeführt werden.



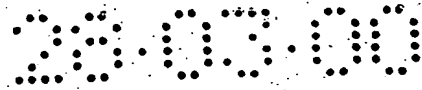
Das logische Diagramm 250 kann auf einem kommerziellen CAD-Paket, wie beispielsweise AutoCAD, erzeugt werden, bei dem es sich um ein gut bekanntes kommerzielles CAD-Paket von AutoDesk, Inc., Sausalito, Kalifornien, handelt. Solche CAD-Pakete, die auf standardmäßigen Rechnern vom PC-Typ arbeiten, dienen zum Erzeugen technischer Zeichnungen für die Entwicklung und Konstruktion verschiedenartiger Erzeugnisse und liefern Datendateien, die die grafische Struktur der Zeichnungen darstellen, welche auf dem elektronischen Sichtschirm des Rechners dargestellt werden. Einige dieser Datendateien werden weiter unten beschrieben.

In Übereinstimmung mit üblicher Praxis bei elektrischen schematischen Darstellungen werden auf der linken Seite des Sichtschirms 255 des Rechners Zeilennummern 256 angegeben. Im gezeigten Fall bilden die Nummern 101 bis 112 den vertikalen linken Rand. Die schematische Darstellung selbst wird interaktiv durch den Programmierer zwischen einer linken und rechten vertikal verlaufenden Linie konstruiert, wobei diese vertikal verlaufenden Linien die elektrischen Energieschienen darstellen. Diese Energieschienen sind mit L1 und L2 bezeichnet und sehen eine symbolische Quelle und Senke für den Strom vor, der durch "Drähte" 258, die auf dem Sichtschirm 255 als Linien dargestellt sind, und durch die elementaren Funktionen der Steuerung fließt, die als logische Ikonen 254 dargestellt sind.

Das oben beschriebene Steuerungssystem 300 ist in der schematischen Darstellung durch das logische Diagramm 250 dargestellt, in welchem eine logische Ikone 254, die mit PB102 bezeichnet ist (Darstellung der Drucktaste 312 von Fig. 12), einen ersten Anschluß "1" hat, der mit der ersten Energieschiene L1 verbunden ist, die in Übereinstimmung mit der herkömmlichen Praxis schematischer Darstellungen auf der linken Seite des Schemabildes vertikal verläuft. Der zweite Anschluß "2" von PB102 ist wiederum unmittelbar zu seiner rechten mit einem ersten Anschluß "1" einer logischen Ikone FLT102 verbunden, die in diesem Fall dazu dient, eine Triggerung der übrigen Schaltung durch extrem kurze Aktivierungen zu vermeiden.

Der zweite Anschluß "2" der logischen Ikone FLT102 ist sowohl mit einer logischen Ikone SOL102 (Darstellung des Solenoids 310 in Fig. 12) als auch einer logischen Ikone LT105 (Darstellung der Lampe 313 in Fig. 12) verbunden, die beide parallel zueinander geschaltet sind und deren zweite Anschlüsse mit der zweiten Energieschiene L2 verbunden sind, die auf der linken Seite des Schemabildes 250 vertikal verläuft.





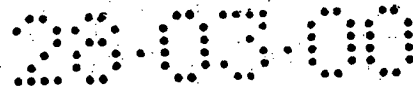
Eine logische Ikone PB320 (Darstellung der Drucktaste 320 in Fig. 12) ist ebenfalls mit ihrem einen Anschluß an die Energieschiene L1 angeschlossen. Ihr zweiter Anschluß ist an einen ersten Anschluß von PRS109 (entspricht dem Näherungsschalter 318 in Fig. 12) angeschlossen. Der Ausgang von PRS109, der normalerweise geöffnet ist, ist jeweils an einen Anschluß von SOL102 (Darstellung des Solenoids 316 in Fig. 12) und von LT112 (Darstellung der Lampe 322 in Fig. 12) angeschlossen.

Der Programmierer konstruiert das logische Diagramm 250 dadurch, daß er logische Ikonen 254 aus einem Menu auswählt und diese mit einem Verdrahtungstool, wie es beispielsweise oben erläutert worden ist, verdrahtet.

Die logischen Ikonen 254 können aus einem logischen Ikonenmenu ausgewählt werden, das auf der rechten Seite des Sichtschirms 255 dargestellt ist und Bilder von standardmäßigen elektrischen schematischen Symbolen zeigt, wie sie in Fig. 5 (a) bis (d) dargestellt sind. Die besonderen logischen Ikonen 254 können durch den Operator durch Bewegung eines Cursors 262 auf dem Sichtschirm 255 ausgewählt werden, und zwar mit Hilfe eines Cursorsteuergeräts, wie beispielsweise einer Maus oder einer Rollkugel. Der Cursor 262 wird über dem geeigneten Symbol des Ikonenmenus 260 positioniert, das dann zu einer geeigneten Position auf dem Schirm "gezogen" wird. Ein Verdrahtungstool (in Fig. 10 gezeigt) kann dann aufgerufen werden, um die verschiedenen einzelnen logischen Ikonen 254 an ihren durch kleine Kreise dargestellten Anschlüssen 264 miteinander zu verbinden.

Jede logische Ikone 254 ist mit einer alphanumerischen Legende 272 versehen, die unterhalb der logischen Ikone 254 dargestellt ist und eine eindeutige Identifizierung 269 für die elementare Funktion vorsieht, die durch diese logische Ikone 254 repräsentiert wird. So ist für die oben erwähnte logische Ikone PB102 die eindeutige Identifizierung 269 einfach "PB102". Die eindeutige Identifizierung 269 wird vom CAD-Programm benutzt, um der logischen Ikone 254 verschiedenartige Eigenschaften der logischen Funktion zuzuordnen, und sie wird automatisch von dem CAD-System erzeugt.

Die eindeutige Identifizierung 269 stimmt mit standardmäßiger schematischer Bild-darstellungspraxis überein, wobei der erste alphanumerische Abschnitt der Legende eine Abkürzung des Gerätetyps ist. Drucktasten verwenden die Buchstaben PB, Näherungsschalter verwenden die Buchstaben PRS, Solenoide die Buchstaben SOL, Lampen die Buchstaben LT



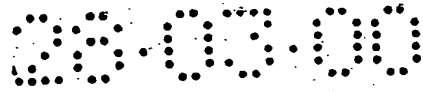
und Filter die Buchstaben FLT. Der numerische Abschnitt der Legende folgt dem alphanumerischen Abschnitt, und es handelt sich hierbei um die Zeilennummer des Geräts, und zwar ausgewählt von den Zeilennummern 256 auf dem Schemabild. Anschlüsse 263 der einzelnen logischen Ikone 254 sind ebenfalls nummeriert, und eine Seitennummer der besonderen Seite des logischen Diagramms 250 kann man bei Schemabildern hinzufügen, die sich über mehrere Seiten erstrecken. Die eindeutige Identifizierung 269 wird dann durch eine interne Datei (nicht gezeigt) mit einem Katalogtyp verknüpft, um für jede durch eine logische Ikone 254 dargestellte elementare Funktion eine zusätzliche Information vorzusehen.

Die Energieschienen L1 und L2 sind ebenfalls als Ikonen (vertikale gerade Linien) dargestellt, sie werden jedoch ohne Eingriff des Bedieners auf dem Sichtschirm automatisch positioniert und benötigen keine weitere Identifizierung.

Drähte 258 sind ebenfalls mit Nummern gekennzeichnet, und zwar bezogen auf ihre Zeilennummern und ihre Reihenfolge in einer Zeile von links nach rechts. Der erste Draht 258 bei der Zeile 102, der mit L1 verbunden ist, hat deshalb die Nummer 1021, der nächste Draht 258 in dieser Zeile die Nummer 1022 und so weiter.

Weil jede logische Ikone 254 aus dem Ikonenmenu 260 ausgewählt wird, erscheint auch auf der rechten Seite des Sichtschirms 255 ein Gerätemenu 266, das verschiedenartige Unterkategorien logischer Ikonen dieser generellen Klasse auflistet und spezifische Katalognummern für kommerzielle Produkte bereitstellt, die in die generelle Klasse der ausgewählten Ikone fallen. In diesem Fall stellen die kommerziellen Produkte einzelne Komponenten dar, die mit einem wahren Draht im Gegensatz zu den virtuellen Komponenten arbeiten, die in dem hochverteilten Steuerungssystem der Erfindung implementiert werden können. Wird beispielsweise eine Drucktastenschalterikone 268 ausgewählt, liefert das Gerätemenu 266 Katalognummern für verschiedene Typen mechanischer Drucktastenschalter mit physischen Klemmschraubenanschlüssen. Diese Drucktasten können normalerweise geöffnet oder normalerweise geschlossen sein und sie können eine oder mehrere Gruppen gekuppelter Kontakte haben, und zwar in Abhängigkeit von der Wahl des Konstrukteurs.

Somit ist jeder einzelnen logischen Ikone 254 eine Katalognummer für ein bestimmtes Gerät zugeordnet. Für viele Katalognummern wird auf dem Sichtschirm 255 dieselbe logische Ikone 254 dargestellt.



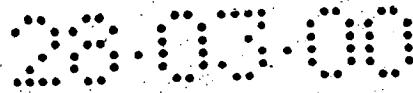
Zusätzliche Information über jede einzelne logische Ikone 154 kann man von der eindeutigen Identifizierung 269 für diese logische Ikone 254 erhalten. Diese zusätzliche Information ist durch eine zweite Datenliste 267 (nicht sichtbar für den Programmierer) vorgesehen, die Katalognummern und eindeutige Identifizierungen der logischen Ikonen 254 mit anderen Daten für die dargestellte elementare Funktion verknüpft. Insbesondere und unter Bezugnahme auf Fig. 16 sieht die zweite Datenliste 267 Spalten 269 und 275 vor, die längs jeweils einer Zeile jede Katalognummer mit einer eindeutigen Identifizierung verknüpfen.

Ein Paar Spalten 271 liefert Information für die benutzte Ikone, und zwar zum Darstellen der elementaren Funktion einschließlich der Anzahl der Eingangs- und Ausgangsanschlüsse, die mit Drähten verbunden werden können.

Eine Spalte 270 sieht einen Zeiger vor, der eine Programmsubroutine identifiziert, die verwendet werden kann, um die durch die eindeutige Identifizierung 269 dargestellte elementare Funktion nachzuahmen. Der Betrieb eines Drucktastenschalters im normalerweise geöffneten oder normalerweise geschlossenen Zustand hat beispielsweise ein Programm von zwei implementierenden Programmen, die in ein Modul geladen werden können, um die Drucktaste nachzuahmen.

Die Datenliste 267 gibt auch eine Klasse 273 der elementaren Funktion an, die im weitesten Sinne diejenigen elementaren Funktionen charakterisiert, welche spezielle Hardware wie Drucktastenoperatoren oder Lampen benötigen. Alle Drucktastenschalter, die einen physischen Drucktastenoperator benötigen, befinden sich in einer einzigen Klasse, in diesem Falle die Klasse A. Das Filter FLT102 das umfassende arithmetische Fähigkeiten benötigt, ist als einer Klasse B zugeordnet dargestellt und deshalb nicht für Module geeignet, die keine übermäßige arithmetischen Fähigkeiten haben. Der Näherungsschalter PRS109, der durch einen einzelnen Näherungsschalter implementiert wird, welcher an Klemmschraubenschlüsse eines allgemeinen Eingangsmoduls angeschlossen wird, gehört einer Klasse C an, wodurch angezeigt wird, daß separate Hardware erforderlich ist.

In einem hohen Maße analog zur Klasse 273 ist eine Anforderungscharakteristik 274, die eine Schätzung der Belastung angibt, welche die elementare Funktion dem implementierenden Prozessor des nachahmenden Moduls aufbürdet. In diesem Fall haben die Drucktastenschalter, die eine relativ einfache Logik haben, eine geringe Anforderung von zehn, wo-



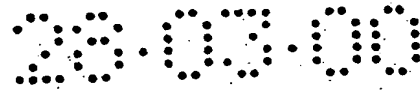
hingegen das Filter, das zusätzliche Prozessorzeit benötigt, eine Anforderung von 110 hat. Mit anderen Worten gibt die Anforderung an, wie hoch die Anforderungen sind, die die Nachahmung an die Hardware stellt.

Eine letzte Datenspalte 276 gibt an, ob für die elementare Funktion eine externe Verbindung notwendig ist. Generell können solche externen Verbindungen Eingänge und/oder Ausgänge von bzw. zu dem gesteuerten Prozeß oder einem Operator sein. Die Relevanz dieser Daten 276 besteht darin, daß diejenigen elementaren Funktionen, die externe Verbindungen haben, nicht ohne weiteres zu anderen räumlichen Stellen verschoben werden können, um das industrielle Steuerungssystem zu optimieren. Bei dem betrachteten Fall hat lediglich die FLT102 keine externe Verbindungen.

Es wird jetzt auf Fig. 13 und 15 Bezug genommen. Das CAD-Programm bewahrt die Daten des logischen Diagramms 250 nicht als Bit-Abbildung des Schemas des logischen Diagramms 250 auf, wie es auf dem Sichtschirm 255 erscheint, sondern vielmehr in Form einer Drahtliste 275, die Drahtnummern der Quelle und dem Ziel logischer Ikonen zuordnet, und zwar wie beschrieben durch ihre eindeutigen Identifizierungen 269 und Anschlußnummern.

Die Drahtliste 275 sieht somit drei Spalten von Daten vor, einschließlich der eindeutigen Drahtnummer und zwei eindeutigen Identifizierungen für die logischen Ikonen 254. Die eindeutigen Identifizierungen werden in diesem Fall durch eine Anschlußnummer ergänzt, die von der eindeutigen Identifizierung 269 durch einen Doppelpunkt getrennt ist. Somit ist PB102:1 der erste Anschluß der elementaren Funktion, der der logischen Ikone PB102 zugeordnet ist. Jede Zeile der Drahtliste 275 identifiziert paarweise Verbindungen zwischen einzelnen logischen Ikonen 254 und dem Draht, der die Verbindung herstellt. Mehrfachzeilen dienen zur Registrierung von Schaltungsverzweigungen.

PB102 von Fig. 12, die direkt mit der linken Energieschiene L1 durch einen Draht 1021 verbunden ist, sieht eine erste Zeile in der Drahtliste 275 als Drahtnummer: 1021 vor, und zwar von L1 nach PB 102:1. Gleichermäßen wo PB102 als nächstes mit FLT103 verbunden ist, der nächste Eintrag in der Drahtliste ist die Drahtnummer: 1022, und zwar von PB102:2 zu SOL 102:1, wodurch aufgezeigt wird, daß der Strom von dem zweiten Anschluß von PB102 zu dem ersten Anschluß von FLT102 fließen kann.



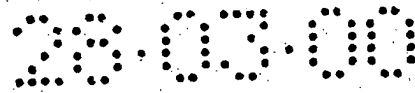
Auf diese Weise sieht die Drahtliste eine Destillation des logischen Schemabildes 250 vor, wobei die Drähte angegeben sind, mit denen die Anschlüsse und einzelnen logischen Ikonen 254 verbunden sind. Die wesentliche Information der Drahtliste wird erzeugt als Datei aus kommerziell verfügbaren Softwaretools, wie beispielsweise dem automatischen Dokumentationssystem PROMIS-E®, hergestellt von ECT, Inc., Brookfield, Wisconsin, das als Anwendung in dem von AutoDesk, Inc., hergestellten AutoCAD-Programm abläuft.

Die Drahtliste 275 definiert zusammen mit der ergänzenden Datenliste 267 vollständig die logische Struktur der Industriesteuerung durch die Zwischenverbindungen von besonderen einzelnen logischen Ikonen 254 mit Drähten 258 und durch die Operation der elementaren Funktionen, die durch die logischen Ikonen 254 dargestellt sind, von denen jede die Operation einer einfachen elektrischen Komponente widerspiegelt.

### **B. Das physische Diagramm**

In der hochverteilten Industriesteuerung 300 sind die elementaren Funktionen, die durch die logischen Ikonen 254 dargestellt sind, vollkommen unabhängig von der physischen Implementation und müssen räumlich separaten Modulen zugeordnet werden, die diese Funktionen implementieren. Die Module, die die elektronische Hardware vorsehen, um die elementaren Funktionen nachzubilden, können Mehrzweckmodule sein. Ein Drucktastenmodul, das einen Drucktastenoperator hat, kann beispielsweise einen normalerweise offenen oder normalerweise geschlossenen Schalter der Erfindung nachbilden und kann Schalter mit Mehrfachkontakten nachbilden, und zwar jeweils ohne irgendeine Änderung in der Hardware. Demnach hat der Programmierer eine beachtliche Flexibilität in der Zuordnung der elementaren Funktionen der einzelnen logischen Ikonen 254 unter einem oder mehreren Modulen.

Anstatt vom Bediener zu verlangen, eine separate Zuordnung von elementaren Funktionen zu Modulen zu verlangen, wird unter Bezugnahme auf Fig. 13 die Zuordnung durch ein zweites CAD-Dokument erreicht, das das physische Layout der Steuerung 300 angibt und physisches Diagramm 252 genannt wird. Ein physisches Diagramm wird normalerweise hergestellt, um die physische Verdrahtung der Steuerung 300 zu gestatten und stellt somit keine neue Konstruktionsaufgabe dar.

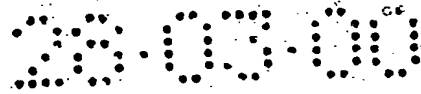


Das physische Diagramm 252 wird auf dem selben CAD-System erzeugt, das zum Erzeugen des logischen Diagramms 250 verwendet wurde, und zwar durch den Bediener unter Anwendung der folgenden Schritte. Zuerst wird ein Menu 356 aufgerufen, das Katalognummern und Beschreibungen von kommerziell erhältlichen Modulen für die elementaren Funktionen vorsieht, wie beispielsweise die Module 352, 350 und 354. Die Module unterscheiden sich in Abhängigkeit ihrer Schnittstelle mit dem gesteuerten Prozeß (beispielsweise durch Anschlüsse, Lampen oder Drucktasten) oder gemäß der speziellen Operation (beispielsweise Transformatoren und Motorsteuerungen). Die Module unterscheiden sich auch gemäß der Fähigkeit ihrer internen Mikroprozessoren und der Menge an Speicherraum im Modul. Bei einer Ausführungsform sieht das Menu 356 lediglich diejenigen Module vor, die verwendet werden können, um die zuvor im logischen Diagramm vorgesehenen elementaren Funktionen zu implementieren. Typischerweise ist mehr als ein Modul vorgesehen, der eine gegebene elementare Funktion implementieren kann, und der Programmierer hat bezüglich der Kombination elementarer Funktionen, die durch ein einziges Modul implementiert werden sollen, hinreichend Flexibilität.

Im Anschluß an die Auswahl eines Moduls erscheint ein rechteckförmiger Umriß 400 auf dem Schirm, und ihm wird eine eindeutige Adresse an der Verbindung 14 zugeordnet, normalerweise in der Reihenfolge der Auswahl. Bei dem dargestellten Beispiel hat die Modulikone 354 die Adresse 7, die Modulikone 350 die Adresse 29 und die Modulikone 352 die Adresse 17.

Es wird jetzt auf Fig. 15 Bezug genommen. Die Verbindung zwischen diesen Modulen und ihren Adressen wird vorgesehen durch eine Moduldatendatei 358 innerhalb des Rechners, die Adressen mit Schirmkoordinaten der Ikonen verknüpft, die die besonderen Module darstellen.

Unter Bezugnahme auf Fig. 17 verknüpft eine zweite Datenliste 360 das Modul durch seine Adresse 282 mit anderen Daten, die das Modul beschreiben und die man vom Menu 356 erhält. Eine Spalte 284 der zweiten Datenliste 360 sieht eine Katalognummer vor, bei der es sich um die Kataloginformation handelt, die im Menu 356 enthalten ist. Diese zusätzliche Information bezüglich der Module wird verwendet, um Kompatibilität zwischen den Modulen und den von den Modulen auszuführenden elementaren Funktionen sicherzustellen, wie es weiter unten noch beschrieben wird.



Ein zweites Menu 362, das ähnlich zu dem Ikonenmenu 260 ist, das in Verbindung mit dem logischen Diagramm 250 beschrieben wurde, sieht eine Ansammlung physischer Ikonen 364 vor, die das äußere Erscheinungsbild der elementaren Funktionen darstellen, die im logischen Diagramm 250 benutzt werden. Generell unterscheiden sich die physischen Ikonen 364 von den logischen Ikonen 254 darin, daß sie das Erscheinungsbild des physischen Geräts anstelle seiner logischen Operation darbieten. Falls das logische Diagramm 250 als erstes repariert wird, sind im Menu 362 nur diejenigen physischen Ikonen 364 dargestellt, welche einzelnen logischen Ikonen 254 im gezeigten logischen Diagramm 250 zugeordnet sind.

Durch die Verwendung des Cursors 262 werden aus dem Menu 362 physische Ikonen 364 ausgewählt und innerhalb der Umrisse der Module 354, 350 und 352 plaziert. So werden beispielsweise zwei "Anschluß"-Ikonen 366, die Ausgänge darstellen, im Umriß plaziert, der das Modul 352 darstellt. Gleichermaßen werden im Umriß, der das Modul 350 darstellt, physische Ikonen 364 für die Lampen 313 und 322 und Drucktasten 312 und 320 plaziert.

Es wird wieder auf Fig. 15 Bezug genommen. Diese Plazierung der physischen Ikonen 364 innerhalb der Modulikonen wird eingefangen in einer Gerätedatendatei 368, die durch das CAD-System konstruiert wird und die die eindeutigen Identifizierungen (beispielsweise PB102), die im logischen Diagramm 250 entwickelt wurden, mit den räumlichen Koordinaten der physischen Symbole auf dem Display gleichsetzt. Die eindeutigen Identifizierungen werden wieder vorgesehen als Legenden 370 im physischen Diagramm 252.

Das physische Diagramm 252 wird vom CAD-System als Moduldatendatei 358 und Gerätedatendatei 368 gespeichert. Die Moduldatendatei 358 verknüpft Adressen auf dem Kommunikationsmittel 14, die einem gegebenen Modul zugeordnet sind, mit den räumlichen Schirmkoordinaten des Umrisses, der dieses Modul darstellt. Gleichermaßen verknüpft die Gerätedatendatei 368 die eindeutige Identifizierung 269 der elementaren Funktionen mit den Schirmkoordinaten ihrer physischen Symbole. Diese Daten zusammen mit einer Identifizierung der physischen Ikone 364, die zur Darstellung der elementaren Funktion der eindeutigen Identifizierung 269 (vorgesehen in der Datenliste 267) verwendet wird, ist hinreichend, um das physische Diagramm zu rekonstruieren und beinhaltet folglich dessen wesentliche Information.

### Kompilierung

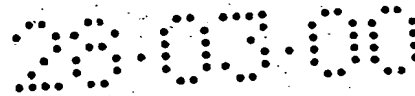
Es wird immer noch auf Fig. 15 und jetzt auch auf Fig. 18 Bezug genommen. Die Erfindung macht Gebrauch von den Dateien, die vom CAD-System erzeugt worden sind und die die Information im logischen und physischen Diagramm 250 und 252 darstellen, um Anweisungen zu erzeugen, die über das Kommunikationsmittel 14 ausgesendet werden, um die verschiedenen Module derart zu programmieren, daß sie sowohl die elementaren Funktionen ausführen und auch virtuell gemäß dem Schema des logischen Diagramms 250 miteinander verbunden sind.

Die Frage, welche Module welche elementaren Funktionen empfangen, wird dadurch gelöst, daß die aus dem physischen Diagramm 252 gewonnene Moduldatendatei 358 und Gerätedatendatei 368 durchgesehen werden, um zu identifizieren, welche elementaren Funktionen bei welchen Modulen aufgrund der Überlappung ihrer Koordinaten auf dem Schirm sind.

Diese durch Prozeßblöcke 402 und 404 in Fig. 18 angezeigte Kompilierung erzeugt eine Verknüpfungstabelle 372, in der jede Zeile vorsieht (1) eine eindeutige Identifizierung 269 für eine besondere elementare Funktion, (2) eine Adresse für dieses Gerät und (3) eine Instanz, wobei die Instanz eine Unteradresse innerhalb der Adresse der Module 350 bis 354 ist und benutzt wird, um diese besondere elementare Funktion zu identifizieren. Gemäß dem physischen Diagramm 252 befindet sich beispielsweise die Drucktaste PB102 im Modul 350 mit der Adresse 29 und der Instanz 3, also die dritte elementare Funktion in diesem Modul. Nachrichten, die an die Adresse 29, Instanz 3 adressiert sind, werden von der Software empfangen, die die Drucktaste PB102 nachbildet.

Als nächstes wird die Tabelle 372 benutzt, um die verschiedenen Module dadurch zu programmieren, daß die Module adressmäßig sortiert werden, diejenigen elementaren Funktionen identifiziert werden, die durch das betreffende Modul nachgeahmt werden, und in der Datenliste 267 der Emulationscode zum Herabladen zu diesem Modul aufgesucht wird. Dieses Herabladen ist angegeben durch Prozeßblöcke 406 in Fig. 18. Im allgemeinen ist der Emulationscode sehr einfach. Für eine Drucktaste liest das Programm einfach Nachrichten, die anzeigen, daß von einer anderen Funktion Strom "empfangen" worden ist, und es sendet





Nachrichten aus, die anzeigen, daß am Ausgang "Strom" vorliegt, wenn die Drucktaste geschlossen ist, wohingegen sonst angezeigt wird, daß am Ausgang kein Strom vorhanden ist.

Als nächstes muß die virtuelle Verdrahtung zwischen den elementaren Funktionen gemäß dem logischen Diagramm 350 durch Prozeßblöcke 408 in Fig. 18 erstellt werden. Diese virtuelle Verdrahtung macht es erforderlich, daß jede elementare Funktion die Adresse von Ausgängen von anderen elementaren Funktionen, mit denen sie verbunden ist, identifiziert, und sie wird mit Hilfe der Verbindungstabelle 376 erzeugt, die eine Adresse VON und ZU vorsieht, welche eine virtuelle Verbindung zwischen jedem Ausgangsanschluß einer elementaren Funktion und jedem Eingangsanschluß einer elementaren Funktion, mit der sie verbunden ist, anzeigen.

Die Verbindungstabelle 376 wird aus der Drahtliste 275 und der Tabelle 372 erzeugt, wie es durch Prozeßblöcke 410 in Fig. 18 angezeigt ist. Speziell wird jede auf der Drahtliste 275 identifizierte elementare Funktion mit ihrer Adresse und Instanz aus der Tabelle 270 verknüpft, um eine zusammengesetzte Adresse zu erzeugen, einschließlich der Adresse des Moduls 354 gefolgt von der Instanz der elementaren Funktion innerhalb des Moduls gefolgt von einer Anschlußnummer der elementaren Funktion. Die Nummer von Anschlüssen des Geräts ist in der Datenliste 267 vorgesehen und ist Teil der Information, die durch die Katalognummer des Geräts vorgegeben ist.

So hat beispielsweise der Eingangsanschluß der Drucktaste PB102 die Adresse 29/3:1, worin 29 die Adresse ihres Moduls ist, 3 die Instanznummer im Modul ist und 1 die Nummer ihres Anschlusses ist, bei dem es sich um den Eingang handelt, wie es im logischen Diagramm 250 gezeigt ist. Der Draht 1021, der das Gerät L1 mit dem ersten Anschluß der Drucktaste PB102 verbindet, ist in der Verbindungstabelle in einer Zeile enthalten, in der die Adresse VON mit L1 (eine Standardbezeichnung, da L1 in jedem logischen Diagramm 250 vorhanden ist) angegeben ist und in der in einer zweiten Spalte die Adresse ZU mit 29/3:1 angegeben ist.

Die Implementierung der virtuellen Verbindung zwischen den Modulen kann dadurch vorgenommen werden, daß jedes Modul mit einem Adresse-ZU-Eintrag aufgefunden wird und diese Zeile der Verbindungstabelle auf dieses Modul geladen wird, um für dieses Modul ein Verbindungsdiagramm (wie es oben beschrieben worden ist) zu erzeugen, wodurch es ihm

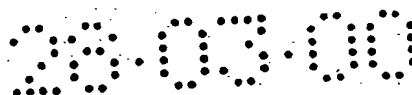


gestattet ist, die Adresse von Nachrichten für seine elementaren Funktionen zu erkennen und diese Nachrichten zu den notwendigen Codefragmenten richtig weiterzuleiten, die auf dem Modul 350 laufen. Dieser Sortiervorgang ist angezeigt in Prozeßblöcken 412 und 414 in Fig. 18.

Es ist verständlich, daß die Erzeugung des physischen Diagramms 252 und des logischen Diagramms 250 mit einer herkömmlichen CAD-Anlage vorgenommen werden kann, die im Rahmen ihrer normalen Betriebsweise Daten produzieren kann, wie sie beispielsweise in der Drahtliste 275, der Moduldatendatei 358 und der Gerätedatendatei 368 vorkommen. Nach der Erzeugung dieser Diagramme braucht sich der Bediener um die Programmierung und Verbindungen zwischen den Modulen keine Gedanken mehr zu machen.

Diese einfache Methode der Programmierung umfaßt in der Praxis auch die Fähigkeit, eine Überprüfung dahingehend durchzuführen, ob die besonderen elementaren Funktionen mit den Modulen, die zu ihrer Emulation ausgewählt worden sind, kompatibel sind. Dies erfolgt unter Verweis auf die Datenlisten 267 und 360, die oben unter Bezugnahme auf Fig. 16 und 17 beschrieben worden sind. Die Spalten geben auch eine Klasse 283 und eine Kapazitätsbewertung 285 an, die generell die Fähigkeiten des Mikroprozessors aufzeigt, der den physischen Modulen 350 bis 354 und spezifischen Hardwareelementen, wie beispielsweise Drucktastenoperatoren und Klemmschraubenanschlüssen zugeordnet ist. Generell muß für eine elementare Funktion, wie sie in der Datenliste 267 beschrieben ist, um in einem gegebenen Modul 350 bis 354 zu arbeiten, die Klasse 273 der elementaren Funktion dieselbe sein wie in der Klasse 283 des Moduls, die in der zweiten Datenliste 360 enthalten ist, und die Gesamtanforderung 274 an Prozessorressourcen von jeder der elementaren Funktionen insgesamt, wie es durch die Datenliste 267 vorgesehen ist, muß kleiner sein als die Kapazität 285 des Moduls, die in der zweiten Datenliste 360 angegeben ist.

Auch unter Bezugnahme auf Fig. 12 hat beispielsweise in der erläuterten Erfindung das Modul 350 mit der Adresse 29 eine Klassenzugehörigkeit von B und eine Kapazität von 100. Sowohl die elementaren Funktionen der Lampen LT105 und LT112 (313 und 322) als auch die Drucktasten PB102 und PB109 (312 und 320) gehören hier der Klasse A an und können deshalb vom Modul 350 ausgeführt werden. Die Gesamtheit von zwei Lampen und zwei Drucktasten haben eine Anforderung von 40 (10 pro elementarer Funktion), die geringer ist als die Kapazität des Moduls 350, so daß auch diese Bedingung erfüllt ist.



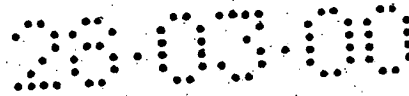
Andererseits hat das Filter FLT102, das zusätzliche Prozessorleistung benötigt, eine Anforderung von 110. Diese Anforderung zusammen mit der Anforderung 20 des Näherungsschalters 109 übertrifft die Kapazität 150 des Moduls 354, so daß dem Bediener ein Fehler signalisiert wird. Das Filter FLT102 hat keine Klasse und ist kompatibel mit irgendeiner Modulkategorie, weil es keine besonderen Anforderungen an externer Hardware hat.

Der Prozeß des Erfassens solcher Inkompatibilitäten zwischen elementaren Funktionen und den zugeordneten Modulen umfaßt eine kreuzweise Bezugnahme jeder elementaren Funktion bezüglich ihres Moduls durch Feststellen, ob das physische Symbol der elementaren Funktion sich innerhalb der Umrisse des Rechtecks befindet, das das besondere Modul darstellt, wie es oben beschrieben worden ist. Die Anforderungen 274 der elementaren Funktionen werden dann mit den Kapazitäten 285 der zugeordneten Module verglichen. Gleichmaßen werden Klassen elementarer Funktionen und Module verglichen.

Konflikte werden dem Bediener auf dem Schirm mit Nachrichten signalisiert, die angeben, daß entweder das ausgewählte besondere Modul die Funktion nicht ausführen kann oder daß zuviele Funktionen von dem besonderen Modul ausgeführt werden sollen, wobei dies während der Erzeugung des physischen Diagramms 252 geschieht. Auf diese Weise kann der Bediener die elementaren Funktionen unter den physischen Modulen unabhängig von den logischen Verbindungen dieser Module im logischen Diagramm 250 neu zuordnen.

Alternativ kann eine automatische Neuordnung der elementaren Funktionen unter den Modulen erfolgen. Als erstes muß jede der elementaren Funktionen dahingehend identifiziert werden, ob sie äußere Verbindungen hat oder nicht. Unter Bezugnahme auf Fig. 16 gibt die ergänzende Datenliste 267 an, daß Drucktastenlampen und ein Näherungsschalter jeweils äußere Verbindungen haben, wohingegen das Filter FLT102 keine hat. Generell ist es nicht praktisch, elementare Funktionen, die äußere Verbindungen haben, zu verschieben, und zwar wegen des zusätzlichen Netzwerkverkehrs, der benötigt wird, um Eingänge und Ausgänge für diese elementare Funktion auf ein entferntes Modul zu übertragen. Dementsprechend führt das Programm die Neuordnung des Filters FLT102 bezüglich des vom Bediener ausgewählten physischen Platzes gemäß dem physischen Diagramm 252 durch.

Diese Neuordnung ist vorzugsweise für den Bediener nicht sichtbar und wird lediglich vom Rechner ausgeführt, der die verfügbaren Module, die in der zweiten Datenliste 360



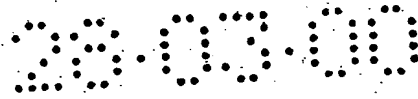
aufgelistet sind, abtastet, um diejenigen Module zu identifizieren, die die geeignete Klasse haben, und der dann überprüft, ob die Hinzufügung des Filters die Kapazität der ausgewählten Module überlasten würde. Das erste Modul, das die Kriterien erfüllt, wird ausgewählt, und das Filter FLT102 wird zu diesem Modul bewegt. Diese Bewegung bedeutet nur, daß die Adresse der elementaren Funktionen, die das Filter FLT102 repräsentieren, gemäß der Adresse des neuen Moduls geändert wird und daß das neue Modul das Programmfragment empfängt, das notwendig ist, um das Filter FLT102 zu implementieren, wenn das hochverteilte System vom Programmiergerät herabgeladen wird.

Falls es erwünscht ist, diese Änderung dem Bediener anzuzeigen, kann die Gerätedatendatei 368 geändert werden, so daß die Koordinate des Filters FLT102 in eine Koordinate innerhalb der neuen Modulikone modifiziert wird, die für die Filterfunktion ausgewählt worden ist.

Somit kann ein herkömmliches CAD-System umfänglich als die Programmiersprache für das hochverteilte System verwendet werden, was gleichzeitig mit dem Vorteil verbunden ist, daß eine akkurate Dokumentation erstellt wird, die sowohl die Logik des industriellen Steuerungssystems als auch das physische Layout seiner Module aufzeigt, wobei das Letztgenannte verwendet werden kann, um die Module tatsächlich miteinander zu verdrahten.

Zahlreiche andere Modifikationen und Abänderungen des bevorzugten Ausführungsbeispiels die im Rahmen und Umfang der Erfindung liegen, sind für den Fachmann augenscheinlich.

Die Erfindung ist auch auf ein hochverteiltes industrielles Steuerungssystem gerichtet, das eine Anzahl separater Steuerungsmodule verwendet, die miteinander über ein gemeinsames Kommunikationsmittel kommunizieren. Jedes Modul ahmt ein oder mehrere grundsätzliche elektrische Teile mit elektrischen Anschlüssen nach, wie beispielsweise Schalter und Relais, und sendet Produktionsnachrichten aus, die den Zustand der Teile anzeigen, beispielsweise, ob ein Stromfluß vorhanden ist oder nicht. Eine Verbindungsliste für jedes Teil in jedem Steuerungsmodul definiert Nachrichtenidentifizierungen von anderen Teilen, deren Produktionsnachrichten von dem Steuerungsmodul empfangen und als Stromfluß zu einem oder mehreren seiner Teile interpretiert werden. Bei einem bevorzugten Ausführungsbeispiel wird das Steuerungssystem in der oben beschriebenen Weise programmiert. Dadurch wird das Pro-



gramm aufgeteilt, so daß es gleichzeitig auf den verschiedenen Modulen laufen kann, ohne daß der Programmierer explizite Anweisungen dahingehend geben muß.

EP 95 107 119.0  
Allen-Bradley Company, Inc.  
91EP

Ansprüche:

1. Verfahren zum Programmieren einer hochverteilten Industriesteuerung aufweisend eine Vielzahl physisch getrennter Module, die über ein gemeinsames Kommunikationsmittel kommunizieren, wobei die Module Nachrichten an das Kommunikationsmittel aussenden und von ihm empfangen, welches Verfahren zum Programmieren die Schritte enthält:
  - a. Generieren eines logischen Diagramms (250), das die Industriesteuerung (300) in der Form eines elektrischen schematischen Bildes aus elektrischen Symbolen auf einem elektronischen Displayschirm darstellt,
    - welche elektrische Symbole (254) elementare elektrische Funktionen darstellen, die von der physischen Implementation unabhängig sind und wenigstens einen Eingangsanschluß und wenigstens einen Ausgangsanschluß haben,
    - welche Anschlüsse Punkte darstellen, zwischen denen gemäß der elementaren Funktion Strom fließt, wobei jeder Anschluß einen Zustand hat, der den Stromfluß anzeigt,
    - welche Anschlüsse durch Linien verbunden sind, die Drähte zu Anschlüssen von anderen elektrischen Symbolen darstellen, und wobei
  - ein solches elektrisches Symbol mit einem Energieversorgungssymbol verbunden ist;
  - b. Generieren eines physischen Diagramms (252) auf dem elektronischen Displayschirm, das die physischen Verbindungen (14) zwischen den Modulen (350, 352, 354) der Steuerung anzeigt und die Module darstellende Umrisse (400) zeigt, die physische Symbole (364) umgeben, welche die elementaren Funktionen darstellen, die den Modulen zuzuordnen und von ihnen auszuführen sind;
  - c. Programmieren der Module (350, 352, 354) auf der Basis des logischen und physischen Diagramms (250, 252) zur Emulation der elementaren Funktion der physischen Symbole, die dem Modul durch das physische Diagramm zugeordnet sind; und
  - d. Kompilieren des physischen und logischen Diagramms (250, 252) zum Generieren einer Kommunikationstabelle, die Kommunikationen zwischen den Modulen (350, 352, 354) über die Kommunikationsverbindung (14) so erstellt, daß die elementaren Funktionen, die von einem gegebenen Modul ausgeführt werden, den Zustand ihrer Anschlüsse zu anderen elementaren Funktionen in anderen Modulen kommunizieren können,

als ob die elektrischen Symbole dieser elementaren Funktionen gemäß dem logischen Diagramm (250) miteinander verbunden wären.

2. Verfahren nach Anspruch 1, bei dem jedes Modul eine eindeutige Adresse auf der gemeinsamen Kommunikationsverbindung hat und bei dem die Kompilation des Schritts (d) eine Adressenliste erzeugt, die für jede gegebene elementare Funktion, welche von einem gegebenen Modul ausgeführt wird, die Adresse von anderen Modulen angibt, die andere elementare Funktionen haben, deren elektrische Symbole mit dem elektrischen Symbol der gegebenen elementaren Funktion durch das logische Diagramm verbunden sind.
3. Verfahren nach Anspruch 1, bei dem die physischen Symbole das Aussehen einer physischen Entsprechung der elementaren Funktion haben.
4. Verfahren nach Anspruch 1, bei dem das physische Diagramm die Verbindung zwischen den Modulen durch die gemeinsame Verbindung mittels einer Linie anzeigt, die die Umrisse miteinander verbindet.
5. Verfahren nach Anspruch 1, bei dem der Schritt (d) enthält:
  - i. Beurteilen des physischen Diagramms zum Zuordnen jedes Anschlusses der elektrischen Symbole zu einer Adresse, die aus einer Adresse des Moduls, dem seine elementare Funktion zugeordnet ist, und einer eindeutigen Subadresse innerhalb dieses Moduls gebildet ist;
  - ii. Beurteilen des logischen Diagramms zum Festlegen der Zwischenverbindungen zwischen jedem elektrischen Symbol; und
  - iii. Inbeziehungsetzen der Zwischenverbindung zu den zugeordneten Adressen zum Erzeugen der Kommunikationstabelle.
6. Verfahren nach Anspruch 1, bei dem jedes Modul einen Fähigkeitsindex hat, der diejenigen elementaren Funktionen angibt, welche von diesem Modul ausgeführt werden können, und bei dem die Kompilation des Schritts (d) das physische Diagramm und den Fähigkeitsindex der Module beurteilt, um zu verifizieren, daß die elementaren Funktionen, die jedem Modul zugeordnet sind, von dem Modul ausgeführt werden können, und, falls nicht, eine Anzeige für einen Benutzer vorsieht.

7. Verfahren nach Anspruch 1, bei dem jedes Modul einen Fähigkeitsindex hat, der diejenigen elementaren Funktionen anzeigt, die von diesem Modul effizient ausgeführt werden können, und bei dem die Kompilation des Schritts (d) auch das physische Diagramm beurteilt und die Zuordnung von elementaren Funktionen zu den physischen Modulen ändert, falls die Zuordnung der elementaren Funktionen, die durch das physische Diagramm angezeigt wird, elementare Funktionen Modulen zuordnet, die diese elementaren Funktionen nicht effizient ausführen können.

8. Verfahren nach Anspruch 7 aufweisend die Schritte:

(e) Identifizieren jeder elementaren Funktion als entweder modulabhängig und damit nicht frei verschiebbar zu einem anderen Modul oder als modulunabhängig und Neuordnung der modulunabhängigen elementaren Funktionen zu anderen Modulen als denjenigen, die durch das physische Diagramm angezeigt werden, um sicherzustellen, daß die modulunabhängigen Funktionen Modulen zugeordnet werden, die die elementaren Funktionen effizient ausführen können.

9. Verfahren nach Anspruch 7, bei dem der Fähigkeitsindex eine Funktion der Modulhardware und der Anzahl der dem Modul zugeordneten elementaren Funktionen ist.

10. Verfahren nach Anspruch 1 aufweisend den Schritt:

(e) Identifizieren jeder elementaren Funktion als entweder modulabhängig und damit als zu einem anderen Modul frei verschiebbar oder modulunabhängig und Neuordnung der modulunabhängigen elementaren Funktionen zu anderen Modulen als denjenigen, die durch das physische Diagramm angezeigt werden, um die Notwendigkeit für Kommunikationen zwischen den Modulen über die gemeinsame Kommunikationsverbindung zu reduzieren.

11. Verfahren nach Anspruch 1 aufweisend den zusätzlichen Schritt:

(e) Laden wenigstens eines Teils der Kommunikationstabelle in jeden Modul, so daß dieser Nachrichten zu anderen Modulen initiieren kann, um den Zustand der Anschlüsse der elektrischen Symbole seiner elementaren Funktionen zu den elementaren Funktionen der anderen Module zu kommunizieren.



12. Verfahren nach Anspruch 1, bei dem das physische Diagramm des Schritts (b) dadurch generiert wird, daß in einem Menü aus Modultypen mit vordefinierter Konstruktion elektrische Symbole ausgewählt werden.

13. Verfahren nach Anspruch 1, bei dem die Module Nachrichtenidentifizierungen haben und Werte ausgeben und die von den Nachrichtenidentifizierungen der Module identifizierte Ausgabewerte mittels Nachrichten über das Kommunikationsmittel kommunizieren, welches Verfahren zum Programmieren die Schritte enthält:

(a) Generieren eines schematischen Bildes miteinander verbundener elektrischer graphischer Symbole auf einem elektronischen Displayschirm, wobei die graphischen Symbole Anschlüsse haben, die Punkte darstellen, zwischen denen Strom fließt, die Anschlüsse durch Linien verbunden sind, die Drähte darstellen, und wobei jedes graphische Symbol einem der Module zugeordnet ist und ein solches graphisches Symbol mit einem Energieversorgungssymbol verbunden ist;

(b) Isolieren jedes gegebenen graphischen Symbols in dem schematischen Bild zum Ermitteln anderer graphischer Symbole, die seine unmittelbaren stromaufwärts gelegenen Nachbarn sind, wobei die anderen graphischen Symbole mit dem gegebenen graphischen Symbol durch eine Linie verbunden sind, die von irgendeinem anderen graphischen Symbol nicht unterbrochen ist, und längs einer Bahn aus Verbindungen durch Linien und graphische Symbole verbunden sind, die zu dem Energieversorgungssymbol führen;

(c) Aussenden, an jedes gegebene Modul, einer Liste der Nachrichtenidentifizierungen der Module, die den unmittelbar stromaufwärts gelegenen Nachbarn des graphischen Symbols entsprechen, das dem gegebenen Modul zugeordnet ist; und

(d) Veranlassen, daß die Module selektiv auf Nachrichten von anderen Modulen antworten, die Nachrichtenidentifizierungen auf der Modulverbindungsliste haben, um virtuelle Verbindungen zwischen den Modulen zu definieren.

14. Verfahren zum Programmieren nach Anspruch 13, bei dem der Ausgabewert eines Moduls dem dargestellten Stromfluß an einem Anschluß des zugeordneten graphischen Symbols entspricht.

15. Verfahren zum Programmieren nach Anspruch 13, bei dem im Schritt (d) ein Wechsel im Ausgang des Moduls die Übertragung einer Nachricht auf dem Kommunikationsmittel veranlaßt.
16. Verfahren zum Programmieren nach Anspruch 13 aufweisend die Schritte:
  - (e) Isolieren jedes gegebenen graphischen Symbols in dem schematischen Bild, um diejenigen anderen graphischen Symbole zu ermitteln, die seine unmittelbaren stromabwärts gelegenen Nachbarn sind, wobei die anderen graphischen Symbole mit dem gegebenen graphischen Symbol über eine Linie verbunden sind, die durch ein anderes graphisches Symbol nicht unterbrochen ist, und längs einer Bahn aus Verbindungen über Linien und graphische Symbole verbunden sind, die zu einem Energieversorgungsrückführsymbol führen; und
  - (f) Aussenden, an jedes gegebene Modul, einer Verbindungsliste aus Nachrichtenidentifizierungen der Module, die den unmittelbar stromabwärts gelegenen Nachbarn des graphischen Symbols entsprechen, das dem gegebenen Modul entspricht.
17. Verfahren zum Programmieren nach Anspruch 13 aufweisend die Schritte:
  - (e) Veranlassen, daß die Module selektiv auf eine verbindliche Berichtsnachricht der unmittelbar stromaufwärts gelegenen Nachbarn der Module antworten, um eine Nachricht auszusenden, die den Moduleausgabewert angibt; und
  - (f) Modifizieren der graphischen Symbole auf dem elektronischen Displayschirm, um jedes Modul anzuzeigen, das auf die verbindliche Berichtsnachricht nicht antwortet.
18. Verfahren zum Programmieren nach Anspruch 17 enthaltend den Schritt des Aussendens einer ersten verbindlichen Berichtsnachricht zu einem regulären vorbestimmten Intervall.
19. Verfahren zum Programmieren nach Anspruch 17, bei dem der Schritt (f) veranlaßt, daß das graphische Symbol, das irgendeinem nicht berichtenden Modul zugeordnet ist, von dem elektronischen Displayschirm verschwindet.

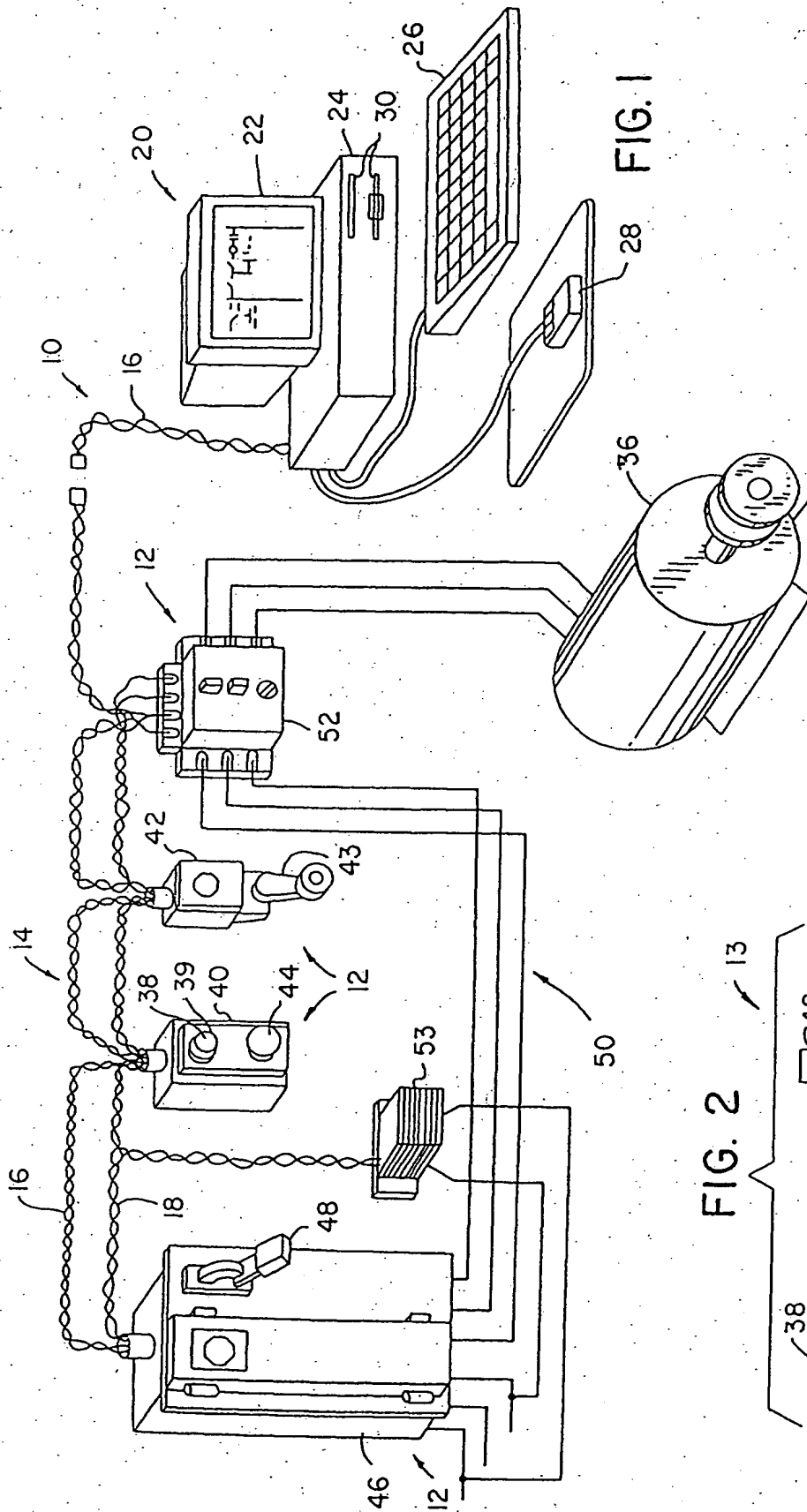


FIG. 1

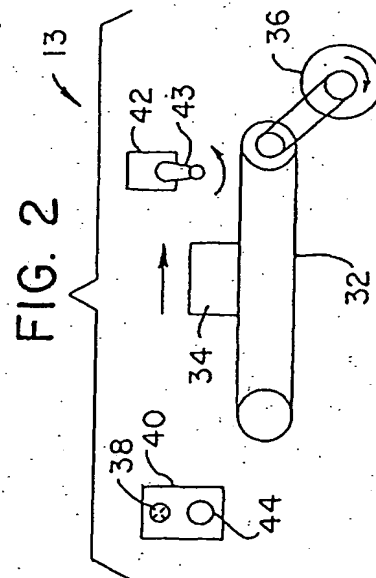
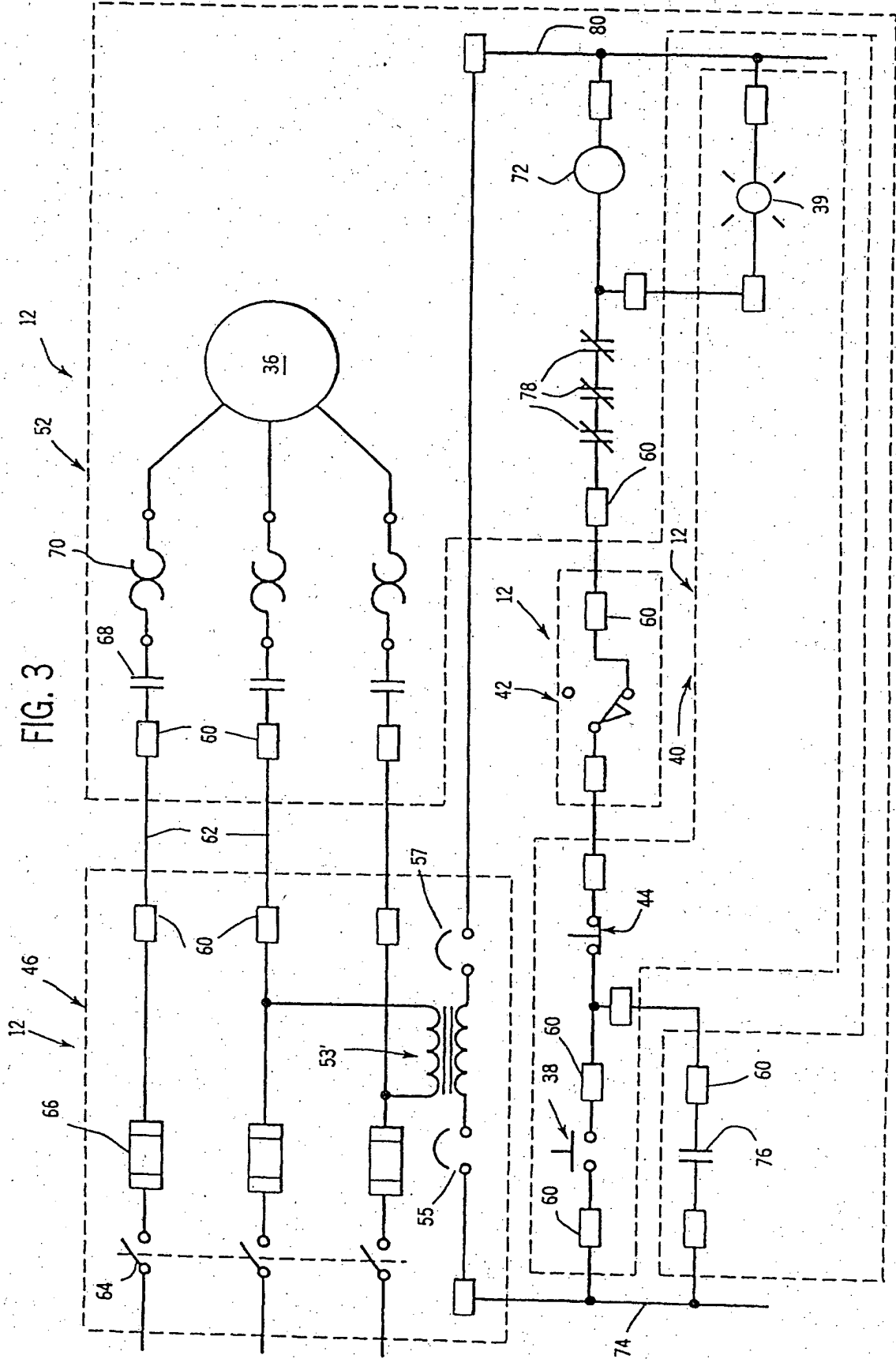


FIG. 2

FIG. 3



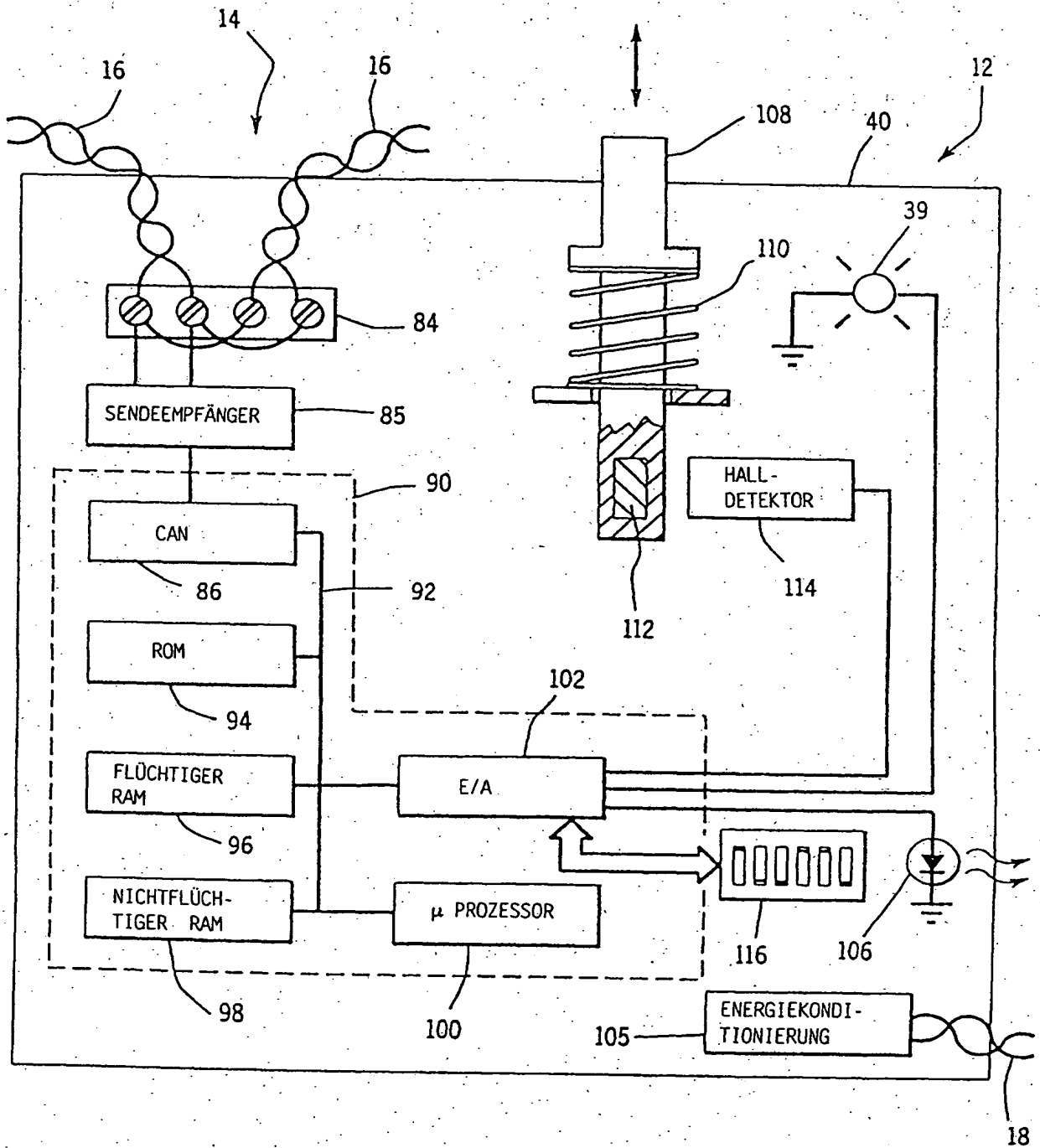


FIG. 4

FIG. 5(a)

TRENNSCHALTER		UNTERBRECHER		LEISTUNGS- SCHALTER		GRENZTASTER					
						ARBEITS- KONTAKT	RUHE- KONTAKT	RUHELAGE NP			
						BETÄTIGT.					
						GESCHLOS- SEN HALTEN		OFFEN- HALTEN			
GRENZTASTER (FORTSETZUNG)						FLÜSSIGKEIT		VAKUUM & DRUCK		TEMPERATUR	
GEHALTENE LAGE	NÄHERUNGSSCHALTER		ARBEITS- KONTAKT	RUHE- KONTAKT	ARBEITS- KONTAKT	RUHE- KONTAKT					
	GESCHLOSSEN	GEÖFFNET									
DURCHFLOSS (LUFT, WASSER, USW.)		FUSS-SCHALTER		KABEL- BETÄTIGTER (NOT-) SCHALTER		STECKER		NICHT-STECKER			
ARBEITS- KONTAKT	RUHE- KONTAKT	ARBEITS- KONTAKT	RUHE- KONTAKT								



FIG. 5(c)

Fig. 300

ZEITVERZÖGERUNG NACH SPULENERREGUNG										RELAIS, USW.		THERMI- SCHE ÜBER- LASTUNG
MASSE	CHASSIS ODER RAHMEN NICHT NOTWENDIGER WEISE GEERDET	STECKER UND BUCHSE	ERREGT			ENTREGT			ARBEITS- KONTAKT	RUHE- KONTAKT		
			ARBEITS- KONTAKT	RUHE- KONTAKT	ARBEITS- KONTAKT	RUHE- KONTAKT						
SPULEN												
RELAIS, ZEITGEBER, USW.	SOLENOIDE, BREMSSEN, USW.				THERMISCHES ÜBERLAST- ELEMENT		STEUERKREIS TRANSFORMATOR					
	ALLGEMEIN	2-STELLUNGEN HYDRAULISCH	3-STELLUNGEN PNEUMATISCH	2-STELLUNGEN ÖL								
SPULEN (FORTSETZUNG)												
AUTOTRANSFORMATOR					LINEARE VARIABLE DIFFERENTIALTRANSFORMATOR					VARIABLE AUTOTRANSFORMATOR		



FIG. 5(d)

DROSSELSPULEN			
SÄTTIGBARER TRANSFORMATOR	SÄTTIGBARER KERN	EISENKERN	SÄTTIGBARER KERN
SPULEN (FORTSETZUNG)			
DROSSELSPULEN (FORTSETZUNG)			
EINSTELLBARER EISENKERN	LUFTKERN	MAGNETVERSTÄRKER-WICKLUNG	3-PHASEN-MOTOR
MOTORE (FORTSETZUNG)			
WIDERSTÄNDE, KONDENSATOREN, USW.			
GLEICHSTROM-MOTOR FELD	WIDERSTAND	HEIZELEMENT	ANGEZAPFTER WIDERSTAND
POTENTIOMETER	RHEOSTAT	3-PHASEN-MOTOR	GLEICHSTROM-MOTOR

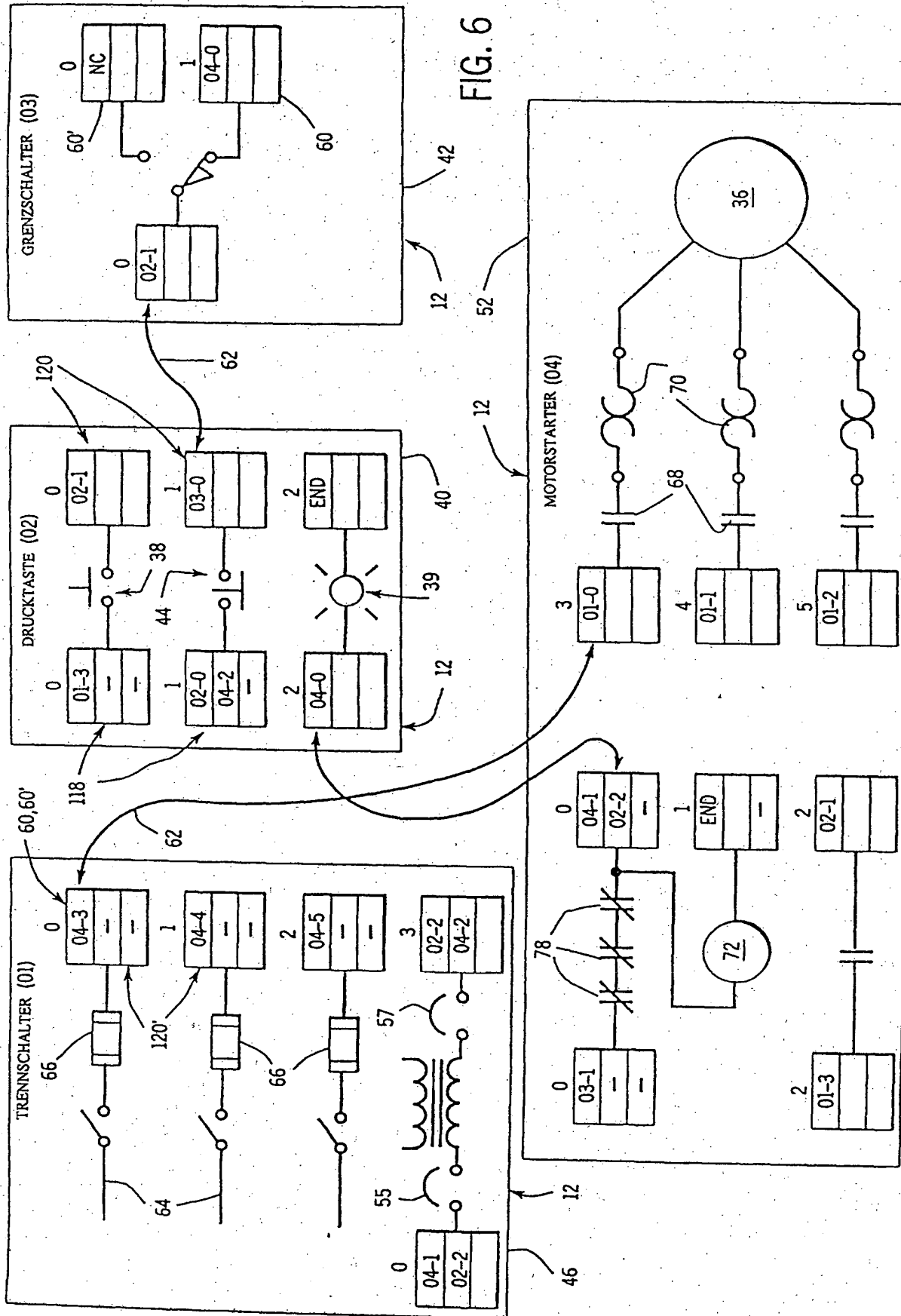


FIG. 6

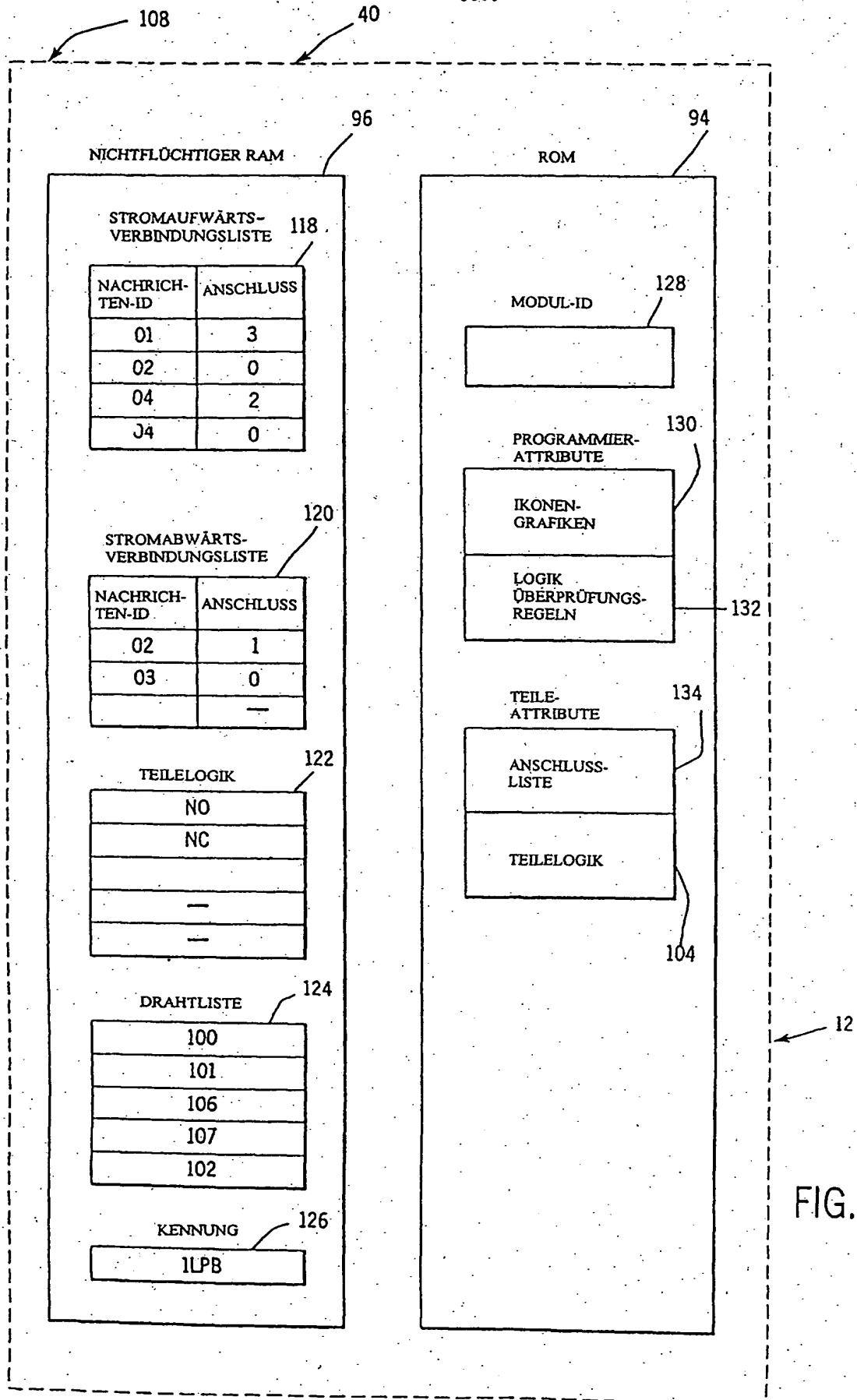


FIG. 7

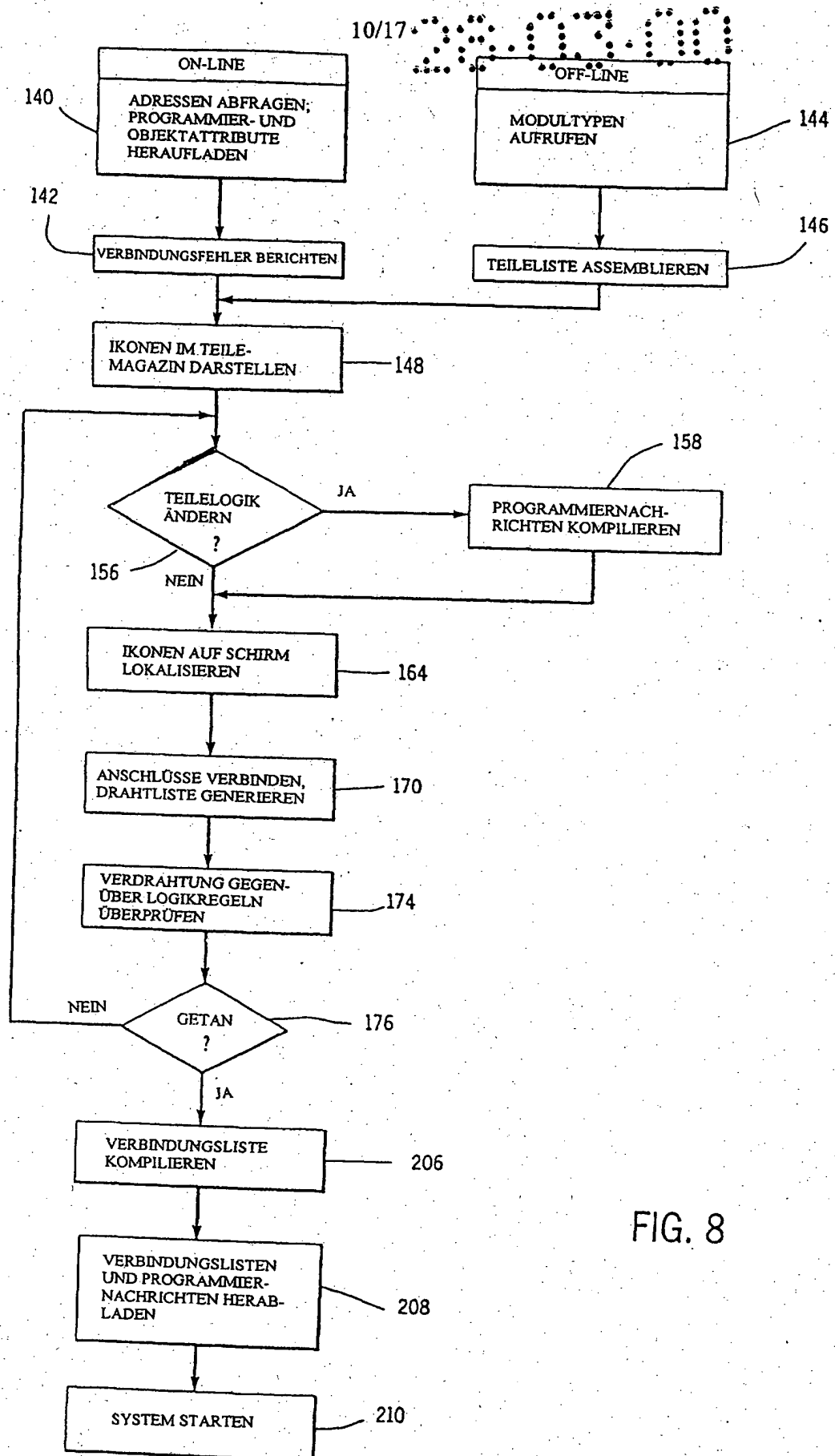
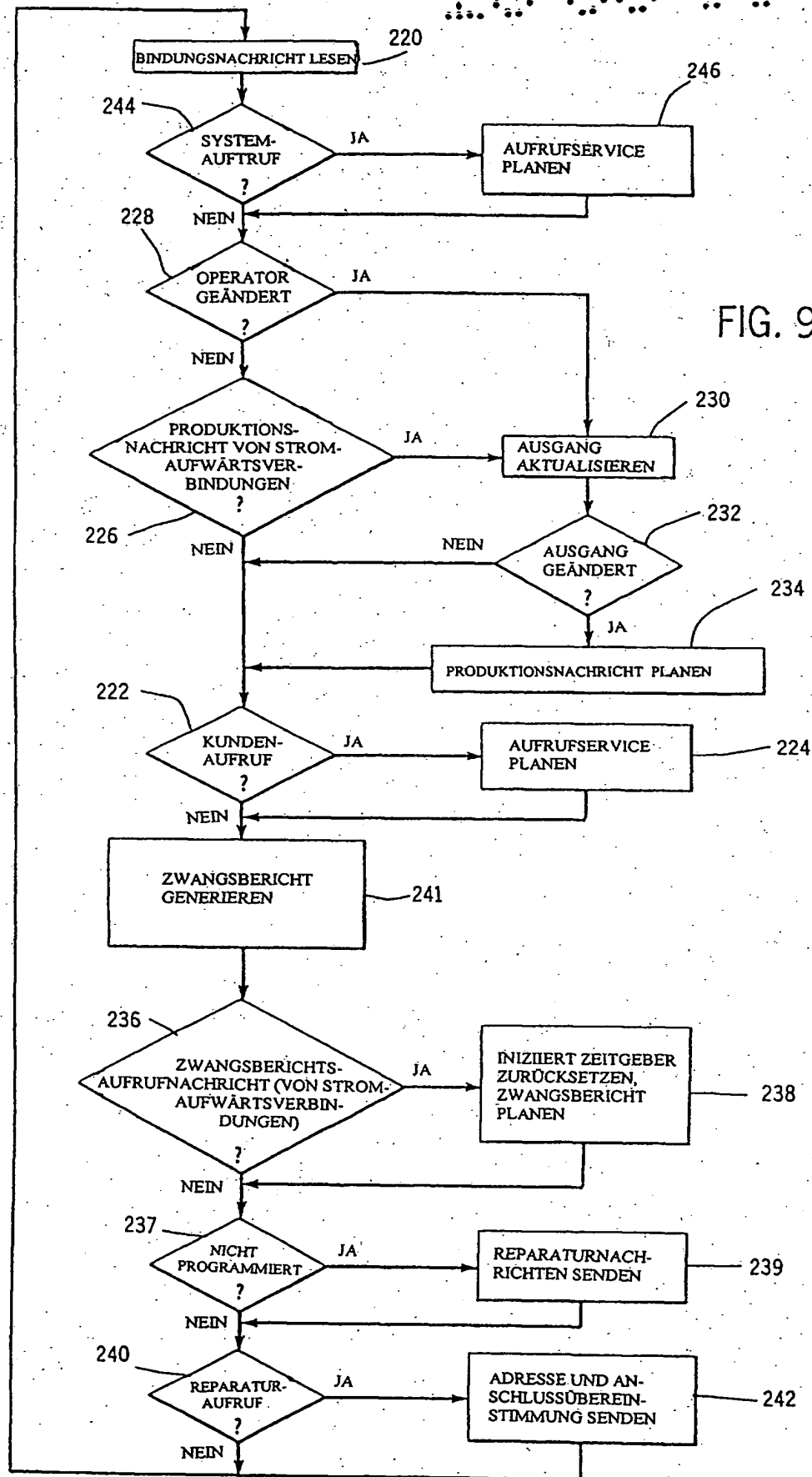


FIG. 8



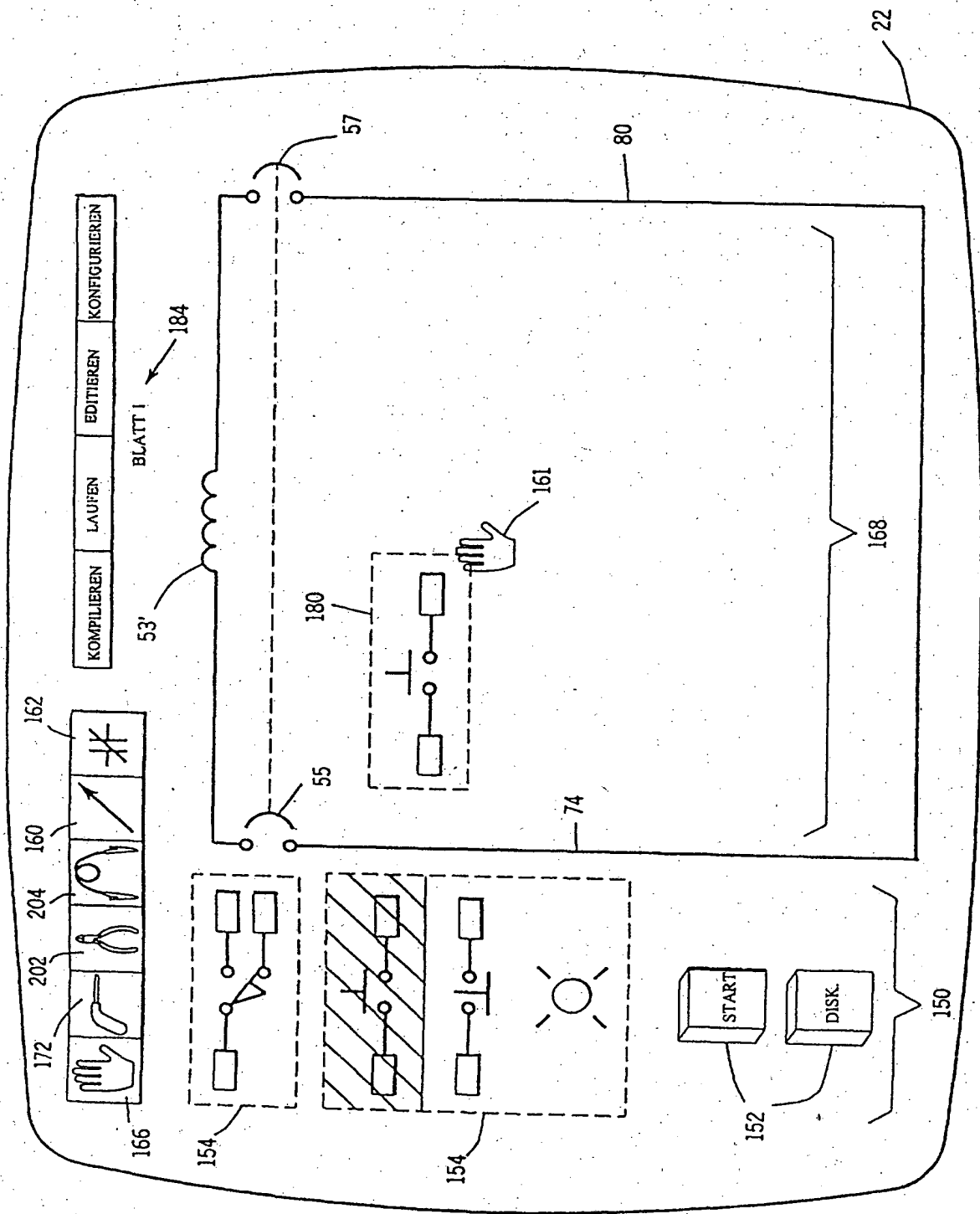


FIG. 10

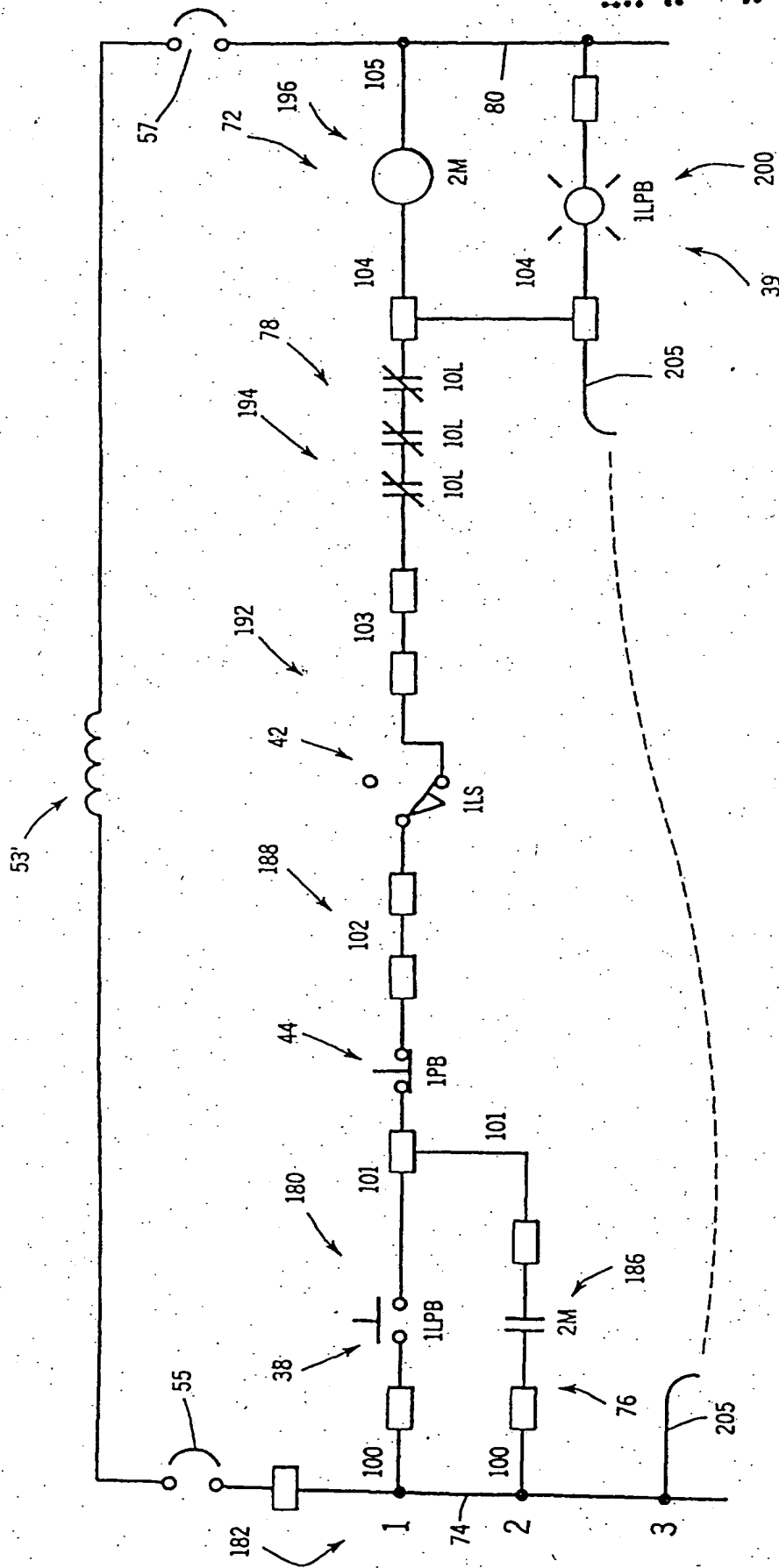
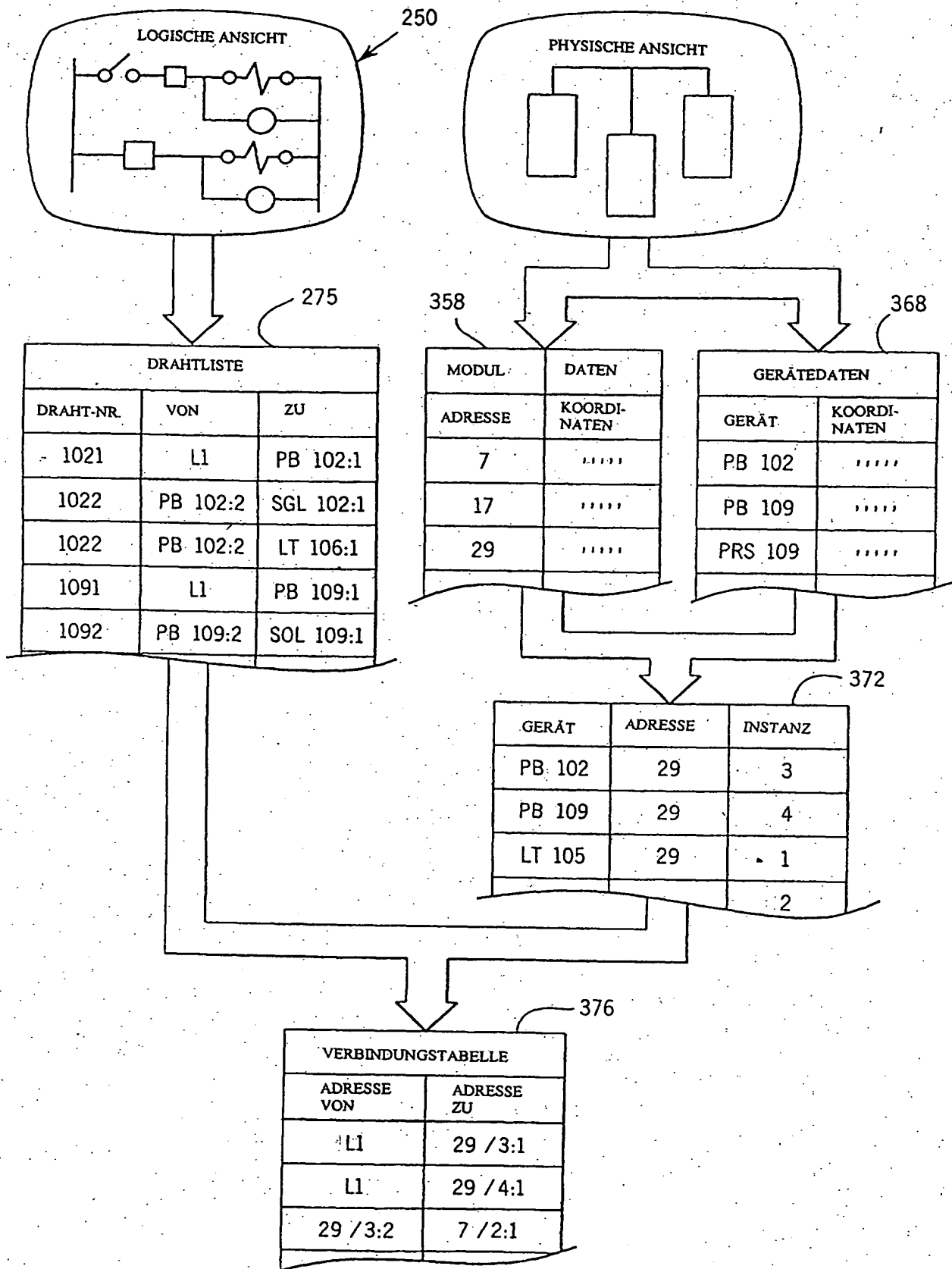


FIG. 11





FIG. 15



16/17 271 273 274 275 276 267

IDENTIFIZIERUNG	EMULATIONS-CODE	IKONE	TERMINAL-ANSCHLUSS	KLASSE	ANFORDERUNG	KATEGORIE	EXTERNE VERBINDUNG
PB 102	ADRESSE 1	XX	1.1	A	10	XXX	JA
LT 105	ADRESSE 2	XX	1.1	A	10	XXX	JA
FLT 102	ADRESSE 3	XX	1.1	-	110	XXX	NEIN
PRS 109	ADRESSE 4	XX	1.1	C	10	XXX	JA

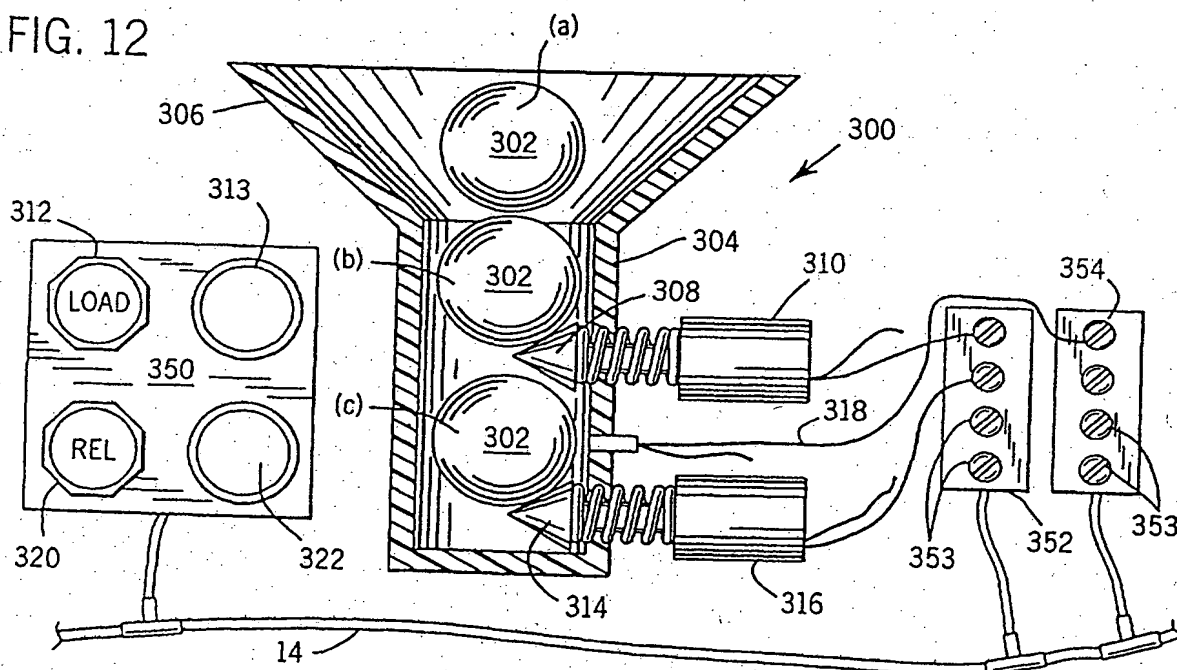
FIG. 16

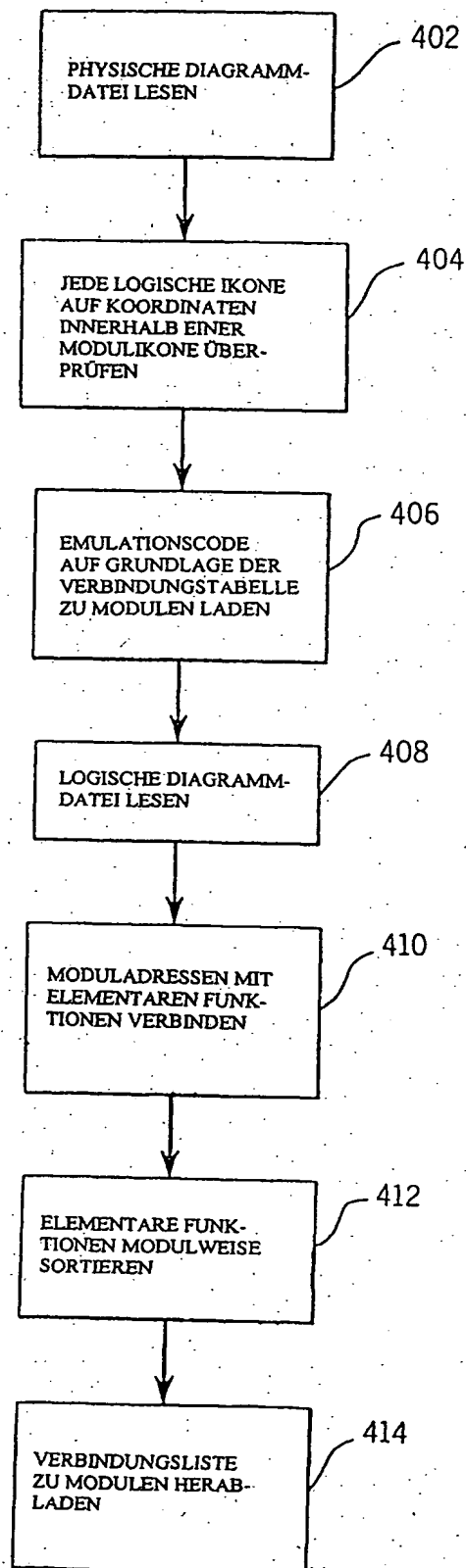
282 283 284 285 360

BEHÄLTER-ADRESSE	KLASSE	KATEGORIE	KAPAZITÄT
7	A	XXX	100
17	B	XXX	150
29	A	XXX	150

FIG. 17

FIG. 12





**THIS PAGE BLANK (USPTO)**